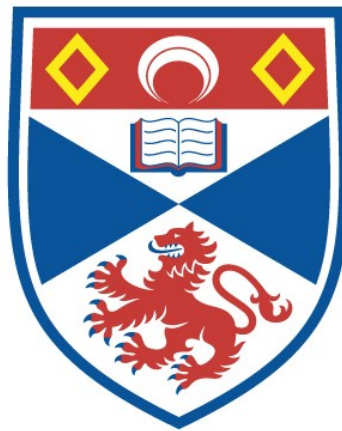


DEFENCE AGAINST DENIAL OF SERVICE (DOS) ATTACKS USING
IDENTIFIER-LOCATOR NETWORK PROTOCOL (ILNP) AND
DOMAIN NAME SYSTEM (DNS)

Khawar Shehzad

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



2019

Full metadata for this thesis is available in
St Andrews Research Repository
at:

<http://research-repository.st-andrews.ac.uk/>

Please use this identifier to cite or link to this thesis:

<http://hdl.handle.net/10023/17833>

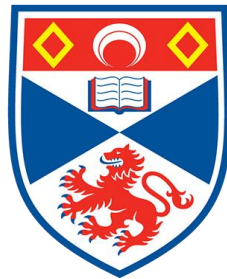
This item is protected by original copyright

This item is licensed under a
Creative Commons License

<https://creativecommons.org/licenses/by-nc-nd/4.0>

Defence against Denial of Service (DoS) attacks using Identifier-Locator Network Protocol (ILNP) and Domain Name System (DNS)

Khawar Shehzad



University of
St Andrews

This thesis is submitted in partial fulfilment for the degree of
Doctor of Philosophy (PhD)
at the University of St Andrews

October 2018

Candidate's declaration

I, Khawar Shehzad, do hereby certify that this thesis, submitted for the degree of PhD, which is approximately 46,400 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for any degree.

I was admitted as a research student at the University of St Andrews in October 2014.

I received funding from an organisation or institution and have acknowledged the funder(s) in the full text of my thesis.

Date 08 April 2019

Signature of candidate

Supervisor's declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date 08 April 2019

Signature of supervisor

Permission for publication

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand, unless exempt by an award of an embargo as requested below, that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that this thesis will be electronically accessible for personal or research use and that the library has the right to migrate this thesis into new electronic forms as required to ensure continued access to the thesis.

I, Khawar Shehzad, confirm that my thesis does not contain any third-party material that requires copyright clearance.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

Printed copy

No embargo on print copy.

Electronic copy

No embargo on electronic copy.

Date 08 April 2019

Signature of candidate

Date 08 April 2019

Signature of supervisor

Underpinning Research Data or Digital Outputs

Candidate's declaration

I, Khawar Shehzad, hereby certify that no requirements to deposit original research data or digital outputs apply to this thesis and that, where appropriate, secondary data used have been referenced in the full text of my thesis.

Date 08 April 2019

Signature of candidate

Abstract

This research considered a novel approach to network security by combining a new networking architecture based on the Identifier Locator Network Protocol (ILNP) and the existing Domain Name System (DNS). Specifically, the investigations considered mitigation of network-level and transport-level based Denial of Service (DoS) attacks. The solutions presented for DoS are applicable to secure servers that are visible externally from an enterprise network. DoS was chosen as an area of concern because in recent years DoS has become the most common and hard to defend against attacks.

The novelty of this approach was to consider the way the DNS and ILNP can work together, transparently to the application, within an enterprise scenario. This was achieved by the introduction of a new application-level access control function — the Capability Management System (CMS) — which applies configuration at the application level (DNS data) and network level (ILNP namespaces). CMS provides dynamic, ephemeral identity and location information to clients and servers, in order to effectively partition legitimate traffic from attack traffic. This was achieved without modifying existing network components such as switches and routers and making standard use of existing functions, such as access control lists, and DNS servers, all within a single trust domain that is under the control of the enterprise.

The prime objectives of this research were:

- to defend against DoS attacks with the use of naming and DNS within an enterprise scenario.
- to increase the attacker's effort in launching a successful DoS attack.
- to reduce visibility of vulnerabilities that can be discovered by an attacker by active probing approaches.
- to practically demonstrate the effectiveness of ILNP and DNS working together to provide a solution for DoS mitigation.

The solution methodology is based on the use of network and transport level capabilities, dynamic changes to DNS data, and a Moving Target Defence (MTD) paradigm. There are three solutions presented which use ILNP namespaces. These solutions are referred to as identifier-based,

locator-based, and combined identifier-locator based solutions, respectively. ILNP-based node identity values were used to provide transport-level per-client server capabilities, thereby providing per-client isolation of traffic. ILNP locator values were used to allow a provision of network-level traffic separation for externally accessible enterprise services. Then, the identifier and locator solutions were combined, showing the possibility of protecting the services, with per-client traffic control and topological traffic path separation.

All solutions were site-based solutions and did not require any modification in the core/external network, or the active cooperation of an ISP, therefore limiting the trust domain to the enterprise itself. Experiments were conducted to evaluate all the solutions on a test-bed consisting of off-the-shelf hardware, open-source software, an implementation of the CMS written in C, all running on Linux. The discussion includes considering the efficacy of the solutions, comparisons with existing methods, performance of each solution, and critical analysis highlighting future improvements that could be made.

Acknowledgements

All praise be to ALLAH Almighty, the Gracious, and the Merciful. My humble gratitude to his slave and messenger MOHAMMED (Peace Be Upon Him) whose exalted character is a source of inspiration for all.

To my primary supervisor, PROF. SALEEM BHATTI, for his wisdom, guidance, and support for everything. I thank him for accepting me as his student, mentoring me along the way, and instilling scientific curiosity and passion for knowledge in me.

To DR. GRAHAM KIRBY, my second supervisor, for all the kind suggestions, and help on how to convey technical information in an easy to understand way for general audience.

To the university administration and computing systems team, for their help in every related matter. I also thank Centre for Academic, Professional, and Organizational Development (CAPOD) for providing valuable courses. I also thank Systems Research Group (SRG) members for their support.

I dedicate this work to my parents, ABDUL QAYYUM and FAHMIDA ANWARI. This work would not have been possible without constant prayers and encouragement from them.

To my dearest, treasured, and beloved wife, HAZIFA AKSAR, for her love, care, and support. My wife and son, HANNAN KHAWAR, have been with me, as a source of ambition, in every step of the way. HAZIFA AKSAR has stood with me at every test of the time, and I thank her for everything.

To all other members of my family who have given guidance and help.

I would like to thank DR. BILQEES SHABBIR for her guidance on how to approach a dissertation.

To D. C. Baird who had tremendous impact on my empirical research through his publication "Experimentation: An Introduction To Measurement Theory and Experiment Design", Third Edition.

To Wayne C. Booth, Gregory G. Colomb, and Joseph M. Williams for their guidance through publication "The Craft of Research", Third Edition.

To all my Ph.D. fellows and friends, including DR. D. PHOOMIKI-ATTISAK, DR. CHONLATEE KHORAKHUN, DR. ADEOLA FABOLA, DR. YUCHEN ZHAO, DR. AHSAN SETHI, DR. AMJAD AL TOBI, MR. DAWAND SULAIMAN, and DR. SHYAM REYAL, for discussing research ideas, and sharing fun time together.

I would also like to thank VERISIGN INC. and UNIVERSITY OF ST ANDREWS for providing funding for this work, without which this work would not have been possible.

Glossary

Attack Surface An attack surface is one or more accessible vulnerabilities in a communication system.

Capabilities A capability is a sequence of bits, with small lifetime, which authorizes a client to access a server or its network.

DNS Capabilities A DNS capability is a type of capability that is distributed through DNS.

DoS Attack Vector It is a means through which an attacker can launch a successful DoS attack.

DoS Detection An act of identifying an imminent or ongoing DoS attack.

DoS Mitigation An act of applying a solution to a DoS attack.

Enterprise A commercial or an industrial undertaking, esp. one involving risk; a firm, company, or business [[OED, 2018](#)].

Fast Flux DNS It is a DNS supported mechanism to shift a node from one namespace to another after a short duration for the same fully qualified domain name (FQDN).

Host Mobility An act of changing the mobile host location from one network to another.

Identifier Locator Network protocol It is a host-based network protocol which provides host/network mobility, host/network multihoming, and scalability of the Internet core as a first class functionality.

ILNPv4 An instantiation of ILNP which is inter-operable with IPv4.

ILNPv6 An instantiation of ILNP which is inter-operable with IPv6.

LC64 Defence A form of ILNP based defence that uses L64 (64-bit Locator) namespace values as capabilities to defend a network against Denial of Service (DoS) attacks. Each LC64 capability uses a different L64 value with a path that is different from other LC64 capability values.

LNC64 Defence A form of ILNP based defence that simultaneously uses L64 (64-bit Node Locator) and NID (64-bit Node Identifier) namespace values as capabilities to defend a host and a network against Denial of Service (DoS) attacks.

Locator It is an ILNP namespace that locates a node. It is formed using upper 64-bits of IPv6 address.

MTD It is a form of defence in which a node's configuration is randomized frequently to secure it from Denial of Service attack while still maintaining operational integrity.

Multi-Vector DoS Attacks DoS attacks which use more than one attack vector to launch a successful DoS attack are termed as multi-vector attacks.

Namespace A namespace is a space of all names for a given node name. IPv6 addresses are node/network names with a space of the number of possible IPv6 addresses that can be created through 128-bits. In IPv6, it is termed as an address space. ILNP has two namespaces with 2 to the power of 64 names in each namespace.

NC64 Defence A form of ILNP based defence that uses NID64 namespace values as capabilities to defend a host against Denial of Service (DoS) attacks.

Network Mobility An act of shifting the mobile network from one network to another.

Node Identifier (NID) It is an ILNP namespace that identifies a particular node, whereas a node can be a physical or virtual host or a physical or virtual network. Each element of this namespace is formed using second half of an IPv6 128-bit address.

Acronyms

ACK Acknowledgement.

ACL Access Control List.

AD Administrative Domain.

AH Authentication Header.

AMI Advanced Metering Infrastructure.

API Application Programming Interface.

AS Autonomous System.

BAck Binding Acknowledgement.

BGP Border Gateway Protocol.

BU Binding Update.

CAP Capability.

CAPAX Capital Expenditure.

ccTLD country code TLD.

CERL Computer-based Education Research Laboratory.

CGA Cryptographically Generated Addresses.

CM Capability Mapping.

CMS Capabilities Management System.

CMSD CMS Daemon.

CMSVD CMS Victim Daemon.

CN Correspondent Node.

CoA Care of Address.

DAD Duplicate Address Detection.

DDNS Dynamic DNS.

DDoS Distributed Denial of Service.

DHCP Dynamic Host Configuration Protocol.

DMZ Demilitarized Zone.

DNS Domain Name System.

DNSSEC Domain Name System Security Extensions.

DoS Denial of Service.

DPI Deep Packet Inspection.

E2E End-to-End.

EDNS0 Extension Mechanisms for DNS.

EID Endpoint Identifier.

EIP Evasive Internet Protocol.

ESD End System Designator.

ESP Encapsulating Security Payload.

ETR Egress Tunnel Router.

EUI-64 64-bit Extended Unique Identifier.

FA Foreign Agent.

FHSS Frequency Hopping Spread Sprectrum.

FQDN Fully Qualified Domain Name.

FSM Finite State Machine.

GL Glocal Locator.

GLI-Split Global Locatro, Local Locator, and Identifier Split.

GSE Global, Site, and End-system address elements.

GSO Generic Segmentation Offload.

gTLD generic TLD.

HA Home Agent.

HI Host Identifier.

HIP Host Identity Protocol.

HMIPv6 Hierarchical Mobile IPv6.

HTTP Hyper Text Transport Protocol.

HTTPS HTTP Secure.

IAB Internal Architectural Board.

ICANN The Internet Corporation for Assigned Names and Numbers.

ICMP Internet Control Message Protocol.

ICMPv6 Internet Control Message Protocol version 6.

ID Identifier.

IDS Intrusion Detection System.

IEEE Institute of Electrical and Electronics Engineers.

IEN Internet Experiment Note.

IETF Internet Engineering Task Force.

IKE Internet Key Exchange.

ILCC Identifier Locator Communications Cache.

ILNP Identifier-Locator Network Protocol.

ILNPv6 Identifier Locator Network Protocol version 6.

IL-v Identifier-Locator vector.

IMS IP Multimedia Subsystem.

IoT Internet of Things.

IP Internet Protocol.

IPSec Internet Protocol Security.

IPv4 Internet Protocol version 4.

IPv6 Internet Protocol version 6.

IRTF Internet Research Task Force.

ISP Internet Service Provider.

ITR Ingress Tunnel Router.

IXP Internet eXchange Point.

L Locator.

L64 64-bit Locator.

LAN Local Area Network.

LC64 L64-based Capabilities.

LISP Locator/ID Separation Protocol.

LKDDb Linux Kernel Driver DataBase.

LL Local Locator.

LNC64 L64 and NID64-based Capabilities.

LP Locator Pointer.

LRR Locator Rewriting Relay.

LSR Loose Source Routing.

LTE Long Term Evolution.

LU Locator Update.

LU-ACK Locator Update Acknowledgement.

MAC Media Access Control.

MAG Mobile Access Gateway.

MAN Metropolitan Area Network.

MANET Mobile Adhoc NETwork.

MAP Mapping.

MILSA Mobility and Multihoming Supporting Identifier-Locator Split Architecture.

MIP Mobile-IP.

MIPv4 Mobile-IP version 4.

MIPv6 Mobile-IP version 6.

MN Mobile Node.

MP-TCP Multipath Transmission Control Protocol.

MSS Maximum Segment Size.

MTD Moving Target Defence.

MTU Maximum Transmission Unit.

NAT Network Address Translation.

NC64 NID64-based Capabilities.

ND Neighbour Discovery.

NEMO Network Mobility.

NGN Next Generation Networks.

NID Node IDentity.

NIST National Institute of Standards and Technology.

NPTv6 IPv6-to-IPv6 Network Prefix Translation.

OPEX Operational Expenditure.

OS Operating System.

OSI Open Systems Interconnections.

QoE Quality of Experience.

QoS Quality of Service.

RA Router Advertisement.

RAM Random Access Memory.

RANGI Routing Architecture for the Next Generation Internet.

rDNS Reverse Domain Name System.

RESTful REpresentational State Transfer.

RFC Request For Comments.

RG Routing Goop.

RLOC Routing Locator.

RO Route Optimization.

ROVER Route Origin VERification.

RPKI Resource Public Key Infrastructure Framework.

RR Resource Record.

RRDNS Round-Robin DNS.

RTT Round Trip Time.

SA Security Association.

SACK Selective ACK.

SANE Secure Architecture for the Networked Enterprise.

SBR Site Border Router.

SCTP Stream Control Transmission protocol.

SDN Software Defined Network.

SHIM6 Level 3 Multihoming Shim Protocol for IPv6.

SIP Session Initiation Protocol.

SLAAC Stateless Address Auto Configuration.

SMTP Simple Mail Transfer Protocol.

SOCKS SOCKet Security.

SSH Secure Shell Protocol.

SSL Secure Socket Layer Protocol.

TCB Transmission Control Block.

TCP Transport Control Protocol.

TLD Top Level Domain.

TLS Transport Layer Security.

TSO TCP Segmentation Offload.

TTL Time To Live.

TVA Traffic Validation Architecture.

UDP User Datagram Protocol.

ULID Upper Layer Identifier.

URI Uniform Resource Identifier.

VIP Virtual Internet Protocol.

vLAN virtual Local Area Network.

VNA Virtual Network Address.

VoIP Voice over IP.

WAN Wide Area Network.

WLAN Wireless Local Area Network.

XML Extensible Markup Language.

XMLRPC eXtensible Markup Language - Remote Procedure Call.

Contents

Abstract	iii
Acknowledgements	vi
List of Figures	xxiv
List of Tables	xxviii
1 Introduction	1
1.1 Denial of Service (DoS) Attacks	2
1.1.1 Definition	2
1.1.2 Severity	2
1.1.3 Attack Vector Diversity	3
1.1.4 Ease Of Attack Execution	3
1.1.5 Lack In Deployment Of Mitigations	3
1.2 Reference Enterprise Scenario	4
1.3 Challenges in Enterprise Defence	5
1.3.1 Host Security	5
1.3.2 Network Security	7
1.3.3 Across-The-Board Security	7
1.4 Moving Target Defence (MTD) Mechanism for Attacker En- tropy	8
1.5 DNS Capabilities Mechanism for Host/Network Access Au- thorization	8
1.6 DNS Fast Flux Mechanism For Agility	9
1.7 Thesis Outline	10
1.7.1 Research Motivation and Approach	10
1.7.2 Research Questions	10
1.8 Thesis Structure	11
2 ILNP, DNS, And DoS Attacks	13
2.1 Identifier-Locator Network Protocol (ILNP)	13
2.1.1 ILNP Architecture	14
2.1.1.1 Node Identifier (NID) Namespace	14

2.1.1.2	Locator (L) Namespace	15
2.1.1.3	Solving Address Entanglement In IP Network Stack	15
2.1.2	ILNPv6 — An Engineering Instantiation of ILNP . . .	16
2.1.2.1	Identifier-Locator Communication Cache (ILCC) . . .	17
2.1.3	ILNP Mobility	17
2.1.3.1	Host Mobility	18
2.1.3.2	Site Mobility	19
2.1.4	ILNP And Security	21
2.2	Domain Name System (DNS)	22
2.2.1	DNS Architecture	22
2.2.2	DNS Engineering	23
2.2.2.1	Resolver Implementations	24
2.2.2.2	DNS Transport Protocols And Security . . .	26
2.2.3	DNS And ILNP	27
2.2.3.1	New DNS Resource Records (RRs) for ILNP . . .	27
2.2.3.2	Modifications In The Client Side DNS APIs For ILNP	30
2.3	DoS Attacks	31
2.3.1	History of DoS Attacks	31
2.3.1.1	1974-2000	31
2.3.1.2	2001-2003	32
2.3.1.3	2004-2008	33
2.3.1.4	2009-2013	33
2.3.1.5	2014-2018	34
2.3.2	Attack Vectors And Classification Of DoS Attacks . .	37
2.3.2.1	Low Data-Rate Transport Layer Attacks . .	37
2.3.2.2	Volumetric Attacks	38
2.3.2.3	Packet Fragmentation Attacks	38
2.3.2.4	Application Layer Attacks [Ranjan et al., 2009] . .	38
2.3.3	Vulnerability Testing And Penetration Testing	39
2.3.4	DoS Detection	39
2.3.5	DoS Mitigation	39
2.3.6	Existing Mitigations Against Low Data Rate SYN Floods .	40
2.3.6.1	SYN Cookies	40
2.3.6.2	SYN Caches	41
2.3.7	Existing Mitigations Against Volumetric DoS Attacks .	41
2.3.7.1	Round Robin DNS (RRDNS)	41
2.3.7.2	Reverse Proxy Through A Cloud Provider . .	42
2.3.7.3	In-line Filtering	44
2.3.7.4	Reputation Based Approaches	44
2.3.8	Lack Of Flexible And Elastic DoS Defences	45
2.4	Security Paradigms	45
2.4.1	Moving Target Defence (MTD)	45

2.4.1.1	Moving Property	46
2.4.1.2	Authorization Property	47
2.4.2	Domain Name System (DNS) Capabilities	48
2.4.2.1	Examples Of DNS Capabilities-Based Defences	49
2.4.2.2	Non-Domain Name System (DNS) Based Capabilities	50
2.4.3	Denial Of Capabilities Attacks	52
2.4.3.1	Defence Against Denial Of Capabilities Attacks	52
2.5	Naming Based Protocols And DoS Defence	53
2.5.1	Locator/ID Separation Protocol (LISP) And Distributed Denial of Service (DDoS) Defence	53
2.5.2	Host Identity Protocol (HIP) And DDoS Defence	54
2.5.3	Identifier-Locator Network Protocol (ILNP) And DoS Defence	54
2.6	Security Challenges And Solutions Matrix	54
2.6.1	Network Reconnaissance Attacks	54
2.6.2	ILNP-based Security Solutions	55
2.6.3	The Role Of DoS Fail-over	55
2.7	Summary	55
3	Enterprise Host Defence	57
3.1	Rationale	58
3.2	NID64 Capabilities (NC64)	58
3.2.1	New Network Components In NC64 Defence	59
3.2.1.1	Capabilities Management Server (CMS)	59
3.2.1.2	CMS Module For DNS (MODCMS)	60
3.2.1.3	CMS Victim Daemon (CMSVD)	60
3.2.2	Properties Of A NC64 Capability	60
3.2.3	NC64 Defence Capability Mappings	61
3.2.4	Design For A NC64 Defence Implementation	61
3.2.5	Finite State Machines (FSMs) Of NC64 Defence Components	62
3.2.5.1	FSM for CMS	62
3.2.5.2	FSM for MODCMS	63
3.2.5.3	FSM for CMSVD	64
3.2.6	Defence Protocol	65
3.3	Empirical And Comparative Evaluation Of NID64-based Capabilities (NC64) And SYN Cookies	66
3.3.1	Methodology And Experiment Design	66
3.3.2	Results	71
3.3.2.1	LAN Environment	71
3.3.2.2	MAN Environment	74
3.3.2.3	WAN Environment	76

3.3.3	Statistical Analysis	78
3.3.3.1	LAN Analysis	78
3.3.3.2	MAN Analysis	80
3.3.3.3	WAN Analysis	82
3.3.4	NC64 And SYN Cookies — Performance Similarity And Overall Benefits	83
3.4	Evaluating Performance Of NC64 Distribution	84
3.4.1	Experiment Design	84
3.4.2	Testing	85
3.4.3	Results	86
3.5	Summary	92
4	Enterprise Site Defence	93
4.1	Rationale	94
4.1.1	Definitions And Assumptions	95
4.1.1.1	ILNP Locator-based Capabilities (LC64)	95
4.1.1.2	NC64 vs LC64	95
4.1.1.3	LC64 MODCMS: LC64 CMS Module Within DNS	95
4.1.1.4	LC64 Capabilities Management System (LC64 CMS)	96
4.1.1.5	Assumptions	96
4.2	Empirical Evaluation Of LC64 Defence	96
4.2.1	Defence Protocol	96
4.2.2	Enterprise Network Architecture With New Network Components	98
4.2.3	LC64 Defence Methodology And Experiment Design	102
4.2.4	Results And Statistical Analysis	105
4.3	Quantifying Attacker's Effort Displacement	111
4.3.1	Design Of The Experiment	111
4.3.2	Testing	112
4.3.3	Results	116
4.4	Measuring SYN Flood Packets During L64 Transitions	118
4.4.1	Design Of The Experiment	118
4.4.2	Testing	120
4.4.3	Results	120
4.5	Summary	123
5	Multi-layered Security And Client Privacy	124
5.1	Rationale	125
5.2	Empirical Evaluation Of LNC64 DoS Defence	128
5.2.1	Experiment Design	132
5.2.2	Results	136
5.3	Client Privacy Side Effect Through LNC64/LC64 Mechanism	140

5.3.1	An Alternative To MPTCP ADD_ADDR Based Mechanism	141
5.3.2	Emulation Design	141
5.3.3	Results	143
5.4	Summary	144
6	Conclusion And Future Works	146
6.1	Summary And Contributions	146
6.1.1	New Enterprise Security Architecture	146
6.1.2	Defence Provisioning For Enterprise Hosts Through The NC64 Defence	147
6.1.3	Defence Provisioning For Enterprise Networks Through LC64 Defence	147
6.1.4	Multi-Level Enterprise Defence Provisioning Through LNC64 Defence	148
6.1.5	Performance Measurements Of NC64 Defence In Diverse Environments	148
6.1.6	Performance Evaluation Of End-To-End Capability Distribution	149
6.1.7	Eliminating The Capability Sharing Problem Through NC64 Defence	149
6.1.8	Quantifying LU Overhead In LC64/LNC64 Defences	149
6.1.9	Crisp Separation Of Backend Control Traffic And End-To-End Data Traffic	150
6.1.10	Measuring Leaked Attack Traffic During Network Transitions In LC64 Defence	150
6.1.11	Increasing Attacker's Effort	151
6.1.12	Side Effect: Possibility To Enhance Client Privacy	151
6.2	Discussion	152
6.2.1	Security Of ILNP-Based DNS Capability Defences	152
6.2.2	Resource Consumption Of CMSVD, MODCMS, CMS, And Router Scripts	153
6.2.3	Short Duration DoS Attacks	154
6.2.4	DoS Attacks From Within The Enterprise	154
6.2.5	Distributing Traffic Load Among Upstream Providers In LC64/LNC64	155
6.2.6	Fast-flux Multi-homing	155
6.2.7	Vulnerability Scanning And Penetration Testing	155
6.2.8	ILNP Namespace Spoofing	156
6.2.9	Using Botnets For An Attack	157
6.2.10	Displacement Of Control From DNS To Capabilities Management System (CMS)	158
6.2.11	Defence Against Botnet Structure Creation	158
6.2.12	Attack Planning, And Durable Information Aggregation	158

6.2.13	End-to-end Quality of Service (QoS)	159
6.2.14	Industrializing The CMS-based Backend And Scalability	159
6.2.15	Embedding Intrusion Detection System (IDS) In CMS Backend	159
6.3	Future Works	160
A	Identifier-based Mappings' Specification	162
A.1	Control Message Types	163
A.2	ILNPv6 Specific MAP Advertisement	163
A.3	Map Acknowledgements (MAP ACKs)	164
A.4	Map Identifier (ID)s And Capability IDs	164
A.5	Mapping (MAP) Expirations	165
A.6	Security Of MAP Related Control Messages	165
A.7	Firewall Considerations	166
A.8	Capability Format Considerations	166
	Bibliography	167

List of Figures

1.1	Reference Enterprise Network Diagram	5
2.1	Internet Protocol version 6 (IPv6) address space and Identifier Locator Network Protocol version 6 (ILNPv6) namespaces [Atkinson & Bhatti, 2012b]	16
2.2	ILNP based host mobility using handoff	18
2.3	ILNP based network mobility using handoff	20
2.4	Fully Qualified Domain Names (FQDNs) in the context of DNS hierarchy	23
2.5	End-to-end Recursive DNS name resolution	25
2.6	Iterative DNS name resolution	26
3.1	Required modifications in enterprise network for NC64 defence	59
3.2	The Finite State Machine (FSM) for the CMS software	62
3.3	The Finite State Machine (FSM) for the MODCMS software	63
3.4	The Finite State Machine (FSM) for the CMSVD software . .	64
3.5	NC64 Defence Protocol Sequence Diagram	66
3.6	Logical network diagram for NC64 defence against SYN flood attacks. CMS identity and location is a secret to the DNS and victim servers, so it is shown as a part of the protected network. The DNS and victim server is accessible by external networks, so they are shown in the DMZ. Network emulation is done at the routers which attach the enterprise network to the client and attacker networks. There are 2,000 clients (see §3.3.1) so we show them as stacked boxes within the client network.	68

3.7	A testbed for NC64 based defence against SYN flood attacks. Each entity that is attached to a router is part of a separate network. Each router is an unmodified Linux box with packet forwarding enabled. Each router can add packet delay to emulate MAN and WAN environments based on the needs of an individual sub-experiment. This topology is chosen so as to model a real world enterprise network. Each router is creating a separate network (external or internal). R4 and R5 act as edge routers.	69
3.8	Performance evaluation of client-victim communication comparing lack of defence, SYN Cookies, and NC64 mechanisms. The diamond symbol in the figure shows the mean value. . .	73
3.9	Performance evaluation of client-victim communication comparing lack of defence, SYN Cookies, and NC64 mechanisms. The diamond symbol shows the mean value.	75
3.10	Performance evaluation of client-victim communication comparing lack of defence, SYN Cookies, and NC64 mechanisms. The diamond symbol shows the mean value.	77
3.11	Logical diagram for the performance comparison of NC64 capability-backed DNS response distribution and DNS only response distribution to clients	84
3.12	Average response time measurements (milliseconds) taken for the LAN environment with a 0%, 10%, and 20% emulated packet loss. The graph shows information about the scenarios of measuring DNS requests/responses with and without CMS	87
3.13	Average response time measurements (milliseconds) taken for the MAN environment with a 0%, 10%, and 20% emulated packet loss. The graph shows information about the scenario of measuring DNS requests/responses with and without CMS	89
3.14	Average response time measurements (milliseconds) taken for the WAN environment with a 0%, 10%, and 20% emulated packet loss. The graph shows information about the scenario of measuring DNS requests/replies with and without CMS	91
4.1	LC64 Defence Protocol Sequence Diagram	98
4.2	The reference inter-networked enterprise site for which an LC64 defence is designed	99
4.3	MODCMS Finite state machine	100
4.4	LC64 CMS Finite state machine	100
4.5	The testbed for an LC64 defence	103
4.6	Traffic comparison under DDoS with an LC64 defence in place. The diamond symbol shows the mean of all the iterations.	107
4.7	Quantile-quantile plots for all sub-scenarios in the L64 based defence	108

4.8	Locator Updates (LUs) performance and overhead	110
4.9	Testbed for LC64 defence to measure attacker's effort while doing fast flux on both the redundant link and the link under attack.	113
4.10	Sequence diagram for quantifying attacker's effort. It shows host/network interactions while the LC64 defence is in place. The LC64 CMS fast fluxes between the link under attack (Router 1) and the redundant link (Router 2). RA is the Router Advertisement. $t_x : x \in [0, \infty]$ is the time instance. . .	115
4.11	Boxplot showing displacement in attacker effort	117
4.12	The logical diagram showing topological separation enabled by two uplink routers. SYN flood traffic comes through access router 1. The LC64 CMS controls the access routers using Interface Updates (IUs).	119
4.13	The testbed with topological separation	119
4.14	Time series showing fast flux between two locators and effects on SYN flood traffic	122
5.1	A 128-bit LNC64 value formed using a 64-bit LC64 value and a 64-bit NC64 value	126
5.2	An ILNPv6 Subnet Identification in Comparison to an IPv6 Subnet Identification - Source: RFC 6741 [Atkinson & Bhatti, 2012b]	127
5.3	A matrix showing 64 different hex values formed using 6-bits of L _{ss} . Each hex value represents a subnet. Victim one uses subnets coloured as gold, and victim two uses the subnets shown in a dark grey colour.	127
5.4	A logical diagram of an enterprise with eight upstream links to the Internet using eight routers, LNC64 CMS, CMSVD, MODCMS, DNS, public victim servers, client network, and victim network.	129
5.5	The finite state machine of an LNC64 CMS.	130
5.6	The testbed for evaluating the LNC64 defence. Nine interfaces of router five are connected to nine ports of the switch, providing unique and isolated upstream links. Each interface is configured with a unique L64 value. Router 5 acts as an upstream access router for the victim. The attacker attacks on the interface where the client communication takes place, before enabling the defence. Once the LNC64 defence is enabled, the rest of the eight locators are used by the defence. .	135
5.7	The results comparing the two scenarios with no packet delay. One session is run under the LNC64 defence (top row), and the second session is run without the LNC64 mechanism. It also shows the 5% and 10% emulated packet loss environments.	137

5.8	The results comparing the two scenarios with 25 ms end-to-end delay. One session is run under the LNC64 defence (top row), and the second session is run without the LNC64 mechanism. It also shows the 5% and 10% emulated packet loss environments.	138
5.9	The results comparing the two scenarios with 210 ms end-to-end packet delays. One session is run under the LNC64 defence (top row), and the second session is run without the LNC64 defence. It also shows the 5% and 10% emulated packet loss environments.	139
5.10	A subnet hopping matrix showing the subnet slots used by the enterprise host in our emulation. Each subnet is shown as a hex value of six bits (L _{ss}).	142
5.11	Logical diagram of the enterprise network showing components necessary to enhance client privacy.	142
5.12	The results for the emulation showing the client privacy side effect of an LNC64/LC64 DoS defence. The scenarios one and two are shown at the left and the right side respectively.	144
6.1	Logical diagram of an enterprise network. Each new required element for defences is shown in black.	152
6.2	ILNP namespace spoofing and its effects on TCP connection establishment as done by an off-path attacker.	156
6.3	ILNP namespace spoofing and its effects on TCP connection establishment as done by an on-path attacker.	157

List of Tables

2.1	Use of namespaces and addressing information in ILNP & IP [Atkinson & Bhatti, 2012a]	16
2.2	DNS Resource Records of IP and ILNPv6 along with their definitions, zone file presentation formats, and examples. . . .	29
2.3	Summary of some of the DoS attacks from year 1974 to 2018	36
3.1	Significance testing using Welch’s two-sample t-test examinations of normally distributed results for three test cases within the LAN environment with varying packet loss conditions. Confidence intervals are at 99% level. <i>ppr</i> is packets per run.	78
3.2	Significance testing using Welch’s two-sample t-test examinations of normally distributed results for three test cases within the MAN environment with varying packet loss conditions. Confidence intervals are at 99% level. <i>ppr</i> is packets per run.	80
3.3	Significance testing using Welch’s two-sample t-test examinations of normally distributed results for three test cases within the WAN environment with varying packet loss conditions. Confidence intervals are at 99% level. <i>ppr</i> is packets per run.	82
3.4	Summary of response time measurements taken for DNS-only and DNS with CMS scenarios under different packet loss conditions	92
4.1	This table shows the comparison of baseline with an attack-only scenario. It also shows comparison of the attack-only scenario with the LC64 defence scenario. Each measurement is the client bandwidth in Megabytes per second (MB/s) as measured at the victim host.	108

5.1	Two servers are being allocated with two different sequences of L64 values. CMS allows them to hop from one locator to another. Each server might be using a different L64 value at one particular time.	126
5.2	The traffic comparison for the two scenarios with an average bandwidth of 25 runs in Mbps, where locator nine (shown in yellow) is only used for the client communications in scenario one, whereas the client will shift to other links in scenario two. We have rounded up the numbers to at most 4 significant digits.	140
5.3	An example showing the active lifetimes of 8 locators in a chunk of a single client session.	143
A.1	Message types used in an Identifier-based capability system .	163

Chapter 1

Introduction

The wide adoption of Internet services has sparked an interest in their security. This is partly due to the design of IP which did not incorporate DoS security. DoS attacks have become widespread due to their ease of execution and high impact while most of the enterprises offer mission critical services to its clients. Such services may include banking, shopping, and communications. Enterprises deploy security appliances in their networks and sometimes delegate the responsibility to security service providers while paying a heavy financial cost.

DoS attacks are one of the basic tools which attackers use to deny enterprise services to its external clients. So, enterprises lose client trust, enterprise reputation, and money. DoS attacks are a constant threat since their inception. They are easy to launch, using diverse ways, and employ plethora of easily accessible tools to circumvent service availability of even high profile enterprises.

This research introduces new mechanisms to thwart some common DoS attacks. These mechanisms make use of ILNP whose design can help provide an effective DoS security approach which is not possible through IP. Similarly, our mechanisms make use of the DNS infrastructure which is widely deployed and almost every service depends on it. Our mechanisms will provide a first line of defence for such services as their discovery is dependent mostly on DNS.

What follows is an introduction to DoS attacks, our enterprise scenario, challenges in enterprise defence, and our security building blocks, i.e., ILNP architecture and DNS infrastructure.

1.1 DoS Attacks

1.1.1 Definition

According to the National Institute of Standards and Technology (NIST), DoS attacks are defined as:

Actions that prevent the system from functioning in accordance with its intended purpose. A piece of equipment or entity may be rendered inoperable or forced to operate in a degraded state; operations that depend on timeliness may be delayed. [(NIST), 2001]

This work will mitigate some of such actions.

1.1.2 Severity

DoS attacks are becoming rampant, complex, and hard to mitigate [CD-Networks, 2017], [Verisign, 2018]. Since DoS attacks target availability of a host and a network, it becomes a threat to enterprises which are connected either through an intranet or the Internet. DoS attacks are easy to launch and sustain because technologies required to perform such actions are easily available on the Internet. They require sophisticated mechanisms to tackle, and innovation in detecting system vulnerabilities and creating new attack vectors (see §2.3.2) is faster than their mitigations. DoS attacks affect businesses in terms of financial cost, market reputation, industry value, and service uptime. In response, businesses have to spend resources on complex defences, host/network downtime tolerance, security staff, and non-security staff readiness alike [Bhardwaj et al., 2016, Zargar et al., 2013].

Most enterprise solutions are complex, financially burdensome, ineffective, and lack agility. There is much research or works which help mitigate DoS attacks (see §§2.3.6 and 2.3.7). Most of these research works demonstrate complex solutions but real world implementations demand simplicity, ease of execution, maintenance, and even non-extension based first class solutions.

Over time, attackers become more sophisticated and informed, thereby we have new variants of DoS itself, e.g., Distributed Denial of Service (DDoS). Vulnerabilities in existing systems, defences, and network engineering are being exposed due to old IP protocols whose design did not consider such attacks. Due to these reasons new research is mainly focused on creating protocols and architectures which take lessons from previous mistakes in design.

In this research, an effort has been made to address the challenge of DoS mitigation to provide defences which are simple to implement, efficient in operation, and technically feasible in rolling out. These solutions make use

of existing technologies in cooperation with innovative defence paradigms proposed in recent research.

Our research proposes and investigates three DoS defences which provide an enterprise controlled (no requirements to trust external entities, e.g., upstream Internet service providers, independent security providers, etc) backend processor to dynamically generate DNS responses with new host/network naming information from ILNP. It also provides traffic filtration mechanism to separate a legitimate traffic from an attack traffic.

1.1.3 Attack Vector Diversity

DoS attacks can make use of multiple ways to reduce an enterprise service availability. Cyberattack engineering employs social engineering, protocol and application vulnerability testing, and traffic based volumetric floods of packets in order to achieve service disruption to legitimate clients. We will consider a few attack vectors relevant to our research in §2.3.2, with some example attacks on enterprises in §2.3.1.

1.1.4 Ease Of Attack Execution

A plethora of DoS attack software tools are easily accessible over the internet, e.g., via Github¹, Bitbucket², etc. Similarly, there are cloud based DoS-for-hire organizations, providing BOOTERS [Santanna et al., 2017], STRESSERS, and DDOSERS [Mahadev et al., 2016], where one can buy their services to attack any desired on-line resource. Such attacks involve spoofing of source addresses to introduce anonymity of the attacker, and difficulty in filtering attack traffic. It is achieved using simple packet forging mechanisms available in programming languages, such as C (libpcap³), Python (scapy⁴), or even iperf3⁵. In our research, we employed thcsyn6⁶, and iperf3 to emulate DoS traffic in a controlled environment.

1.1.5 Lack In Deployment Of Mitigations

The principle of *being conservative in what you send and liberal in what you accept* [Shue et al., 2012], or *universal reachability* [Ballani et al., 2005], is strongly being debated in favour of providing security and privacy against current and future multi-vector cyberattacks. This principle along with the lack in enterprise-wide deployment of well-researched architectures [Yaar et al., 2004, Andersen, 2003, Anderson et al., 2004, Argyraki & Cheriton,

¹<https://github.com>

²<https://bitbucket.org>

³<https://github.com/the-tcpdump-group/libpcap>

⁴<https://scapy.net>

⁵<https://iperf.fr>

⁶<https://tools.kali.org/information-gathering/thc-ipv6>

2005b, Handley & Greenhalgh, 2004, Stoica et al., 2002, Ioannidis & Bellovin, 2002], contribute to limited security. The pace of attack adoption and attack execution is far more aggressive in comparison to deployment of DoS defences. We present further sets of solutions (chapter 3, chapter 4, and chapter 5), which demonstrate the feasibility of new novel approaches to DoS defence.

1.2 Reference Enterprise Scenario

This research evaluated DoS defences for an enterprise network. Figure 1.1 shows our reference enterprise network diagram. Our defences are placed within the enterprise network without any modification to the outside network except the Internet access router (Site Border Router (SBR)) which itself is controlled by the enterprise. The client node can access any service within the enterprise network based on network policies. The attacker node either uses SYN flood or UDP flood using spoofed source packets. The attacker can only reach an enterprise network using the Internet.

There is no modification done to the firewalls either in the Demilitarized Zone (DMZ) or the protected network. A DMZ is a part of an enterprise network which exposes public services of the enterprise to the untrusted networks. Our defences make use of a host-based firewall which is placed in the web server, mail server, or any other publicly accessible enterprise host.

We use the term victim in two contexts. First, it is the publicly accessible enterprise victim host/server. And second, it is the enterprise network itself. Our defences can mitigate attacks on either enterprise host, its network, or both, based on particular ILNP namespace(s) that is/are in use as part of that defence.

Attackers and clients are assumed to be outside the enterprise network while the enterprise is only accessible through one or more of its Internet Service Provider (ISP)s.

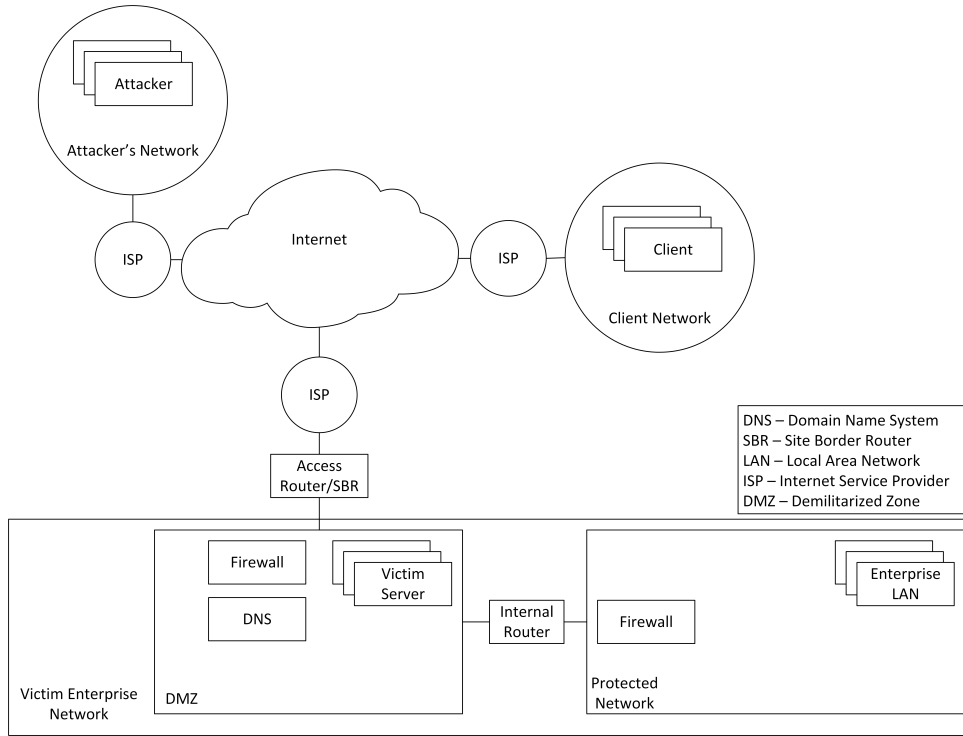


Figure 1.1: Reference Enterprise Network Diagram

1.3 Challenges in Enterprise Defence

1.3.1 Host Security

There are studies about intrusion detection in host security, specifically streamlined towards the application layer [Warrender et al., 1999, Stolfo et al., 2005, Zhang et al., 2004, Zhou et al., 2010]. TCP design makes it vulnerable to attacks as its unauthenticated three-way handshake allows forged header fields. An attacker is able to reach its victims using forged packets which can make its victims unresponsive to new connections. There is a need to device defences which can protect victims from unintended sources. The most common type of attack in such scenarios is a SYN flood, which is the host (an enterprise server that is accessible from the outside enterprise network) challenge addressed in our research.

Approaches to TCP level host security rely on SYN Cookies, SYN Caches, Access Control List (ACL)-based traffic filtration, TCP's Transmission Control Block (TCB) backlog increments, reduction in SYN-RECEIVED Timers, recycling of old half-open connections in TCB, and use of proxies [Eddy, 2007]. But from all the approaches mentioned here, SYN Cookies and SYN Caches are mostly seen as acceptable [Eddy, 2007].

SYN Cookies are lacking in support for TCP packet option fields, whereas SYN Caches do not allow piggybacking of application data over TCP’s three-way handshake. Our objective is to investigate an alternative defence which can prevent such shortcomings in SYN Cookies and SYN Caches for Transport layer TCP SYN floods against TCB exhaustion attacks. SYN Cookies will be part of our evaluation while SYN Caches will be part of our discussion.

Transport Layer Security (TLS)/Secure Socket Layer Protocol (SSL) [Freier et al., 2011], Secure Shell Protocol (SSH) [Ylonen & Lonvick, 2006], and SOCKet Security (SOCKS) [Leech et al., 1996] does provide Transport layer security but their goals are not to mitigate SYN attacks but to support data encryption, client/server Application layer authentication, and data integrity *once* the connections are already setup. This area is thoroughly tested, and mitigations to breaches in integrity and authentication are trusted using TLS/SSL.

Our goal is to provide client authentication at the connection setup level. Our defences make use of TLS/SSL for the integrity of control messages only which are used in interactions of victims with our backend processor. But, our defences do allow flexibility to use any other protocol once an authenticated mechanism is in place before TCP connection establishment.

Host-based security mechanisms rely on TCP protocol modifications, e.g., SYN Cookies change the way a host responds to SYN packets by modifying SYN ACKs, and SYN Caches change the way TCP half-open states of all applications are manipulated in TCBs and global hash tables. These modifications have trade-offs which are not always acceptable [Eddy, 2007]. Our goals are to achieve the same goals of SYN Cookies and SYN Caches but without their respective trade-offs.

There are certain defence approaches which support the perspective that defences must allow end hosts to instruct the network who should be able to initiate connections [Ballani et al., 2005, Handley & Greenhalgh, 2004]. Our solution to host security instructs the victim to configure its firewall, or access control lists, with information on who should be able to initiate the connection. Our solution can, itself, be a subject to a DoS attack. This aspect will also be discussed in the chapters 2, 3 and 6.

In chapter 3, we present an ILNP based defence mechanism that mitigates low rate SYN floods. It has a backend which allows hosts to configure their firewall based on per-client authentication. Our backend derives the DNS responses and the host firewall by allocating a per-client ephemeral naming information that is unique for the duration of an incoming session from the same client. Our backend uses DNS servers as request forwarders to allow interception and modification of individual responses.

1.3.2 Network Security

Enterprise network reachability and its uninterrupted availability is susceptible to volumetric DoS attacks. Network security procedures try to ensure that public hosts running in the network are accessible and the ingress/egress data is authenticated and has integrity. In this work, we will deal with the network accessibility part.

DoS attacks which target network access are termed as traffic based network attacks. Network security makes use of mechanisms which can either be placed at the edge or within the enterprise network. Similarly, there are approaches in which an enterprise requires help from ISPs. We will take the former approach thereby providing defences that are within an enterprise network and within its control.

We see a rise in network-based attacks on mission critical infrastructures of most of the enterprises. Some of such mission-critical infrastructures are Advanced Metering Infrastructure (AMI) smart grid networks, industrial Internet of Things (IoT) networks, industrial control systems [Ylmaz et al., 2018], DNS root servers [Brownlee et al., 2001, Rootops, 2015, ENISA, 2016], Internet eXchange Points (IXPs), fibre-optic backbones, medical systems and military systems.

In 2013, Arbor networks reported DDoS attacks with an average size of 1.77 Gbps (19.5% increase from year 2012) [eSecurity Planet, 2013]. In contrast, we saw an average of 11.2 Gbps in Q1 of 2018 [Verisign, 2018]. In the same quarter, Kaspersky reported that 57.3% of all DDoS attacks were SYN floods [Kupreev; & Badovskaya, 2018].

Some network security defences for DoS make use of edge network appliances (also see §2.3.7.3) which do not scale (in terms of control and data traffic unless more financially burdensome appliances are provisioned) with the intensity of the DDoS attack. Other defences rely on security service providers which handle attack traffic on behalf of an enterprise thereby making all external traffic visible to the security provider (see §2.3.7.2). We provide a solution to DoS attacks against network reachability in chapter 4. We did not test the scalability (in terms of control traffic only) of our solution but it ensures that an enterprise has complete control over defence provisioning and its traffic. We make use of ILNP's topologically significant namespaces along with a backend which controls enterprise DNS and its edge routers. The backend itself is in the protected network of the enterprise. Our solutions are tested to work against UDP and SYN based volumetric DoS attacks.

1.3.3 Across-The-Board Security

An enterprise is required to provide multi-level security to its hosts and networks. It should have mechanisms to thwart attacks on a single host as

well as on the whole network. While ILNP’s non-topologically significant namespace (see §2.1.1.1) provides defence for the host, and ILNP topologically significant namespace (see §2.1.1.2) provides defence for the network; we can combine both approaches or namespaces to provide security at two levels. We investigated this new approach in chapter 5 and found that our prior defences, when combined, can protect an enterprise at both the host and network levels.

While investigating the feasibility of our combined approaches to security, we also noted a side-effect which provides some degree of client privacy. We noted that each client session was distributed over multiple topologically distinct paths with an ephemeral destination namespace which itself changes for the next session from the same client. Thereby making it difficult for an on-path privacy attacker to trace the client session. This is only possible if control signalling in the defence is encrypted §5.3.

1.4 Moving Target Defence (MTD) Mechanism for Attacker Entropy

Moving Target Defence (MTD) is an adaptive security mechanism which obfuscates or intelligently randomizes information elements within hosts and networks to maintain operational integrity of systems [Zhuang et al., 2013]. Such information elements can be derived from hosts and network configurations. MTD tries to achieve randomness and chaos for an attacker but without disrupting end-to-end connectivity for end-hosts.

In our research, we will randomize ILNPv6-based namespaces only for achieving MTD behaviour. The operational integrity of Transport Control Protocol (TCP) connection initiation, path selection, and data communications will be achieved through coupling of MTD and DNS *capabilities* (defined shortly) with the help of ILNPv6’s crisp separation of transport and network layer semantics [Atkinson & Bhatti, 2012a] along with its seamless mobility mechanism [Atkinson et al., 2009a], [Phoomikiattisak, 2016].

1.5 DNS Capabilities Mechanism for Host/Network Access Authorization

A *capability* is an authorization token which is given by a destination to an end-host for a short period of time to reach its network or services running on it.

In our research, we have used ILNPv6 namespaces, i.e., NID64 and L64 values as *capabilities*. The capabilities which use NID values only are termed NID64-based Capabilities (NC64). Each unique NC64 value contains an ephemeral destination identifier and a fixed destination locator (see chap-

ter 3). The capabilities which use L64 values only are termed as L64-based Capabilities (LC64). A unique LC64 value contains an ephemeral destination locator value and a fixed destination identifier (see chapter 4). Similarly, we form a third type of capabilities, termed as L64-NID64-based capabilities (LNC64), which use an ephemeral identifier and an ephemeral locator (see chapter 5). Together, these are called DNS capabilities because DNS is used for their distribution to clients. All of our approaches to DoS defence make use of DNS to distribute ILNP capabilities.

1.6 DNS Fast Flux Mechanism For Agility

DNS fast flux is a network based camouflage mechanism to transition the same domain name from one IP address or ILNP namespace to another within a small duration. DNS host entries are updated frequently to hide the actual attack source.

This is similar to Frequency Hopping Spread Spectrum (FHSS) [Scholtz, 1982] in radio signal transmission in mobile network access technologies where a signal frequently switches a carrier frequency among multiple frequency channels. Anyone who is listening on limited number of carrier frequencies cannot construct the complete transmission.

DNS fast flux is mainly used by hackers to hide harmful activities such as, phishing, malware delivery, covert communications, web proxying etc [Nazario & Holz, 2008]. DNS fast flux can also be used to mitigate volumetric DoS attacks where a particular destination address/namespace of a victim is used. Once an enterprise notices that it is being attacked using a particular destination address/namespace, it can quickly move to other addresses/namespaces. In IPv6, this is not possible unless an enterprise deploys Mobile-IP version 6 (MIPv6) extensions. In IPv4, we require use of NAT [Shue et al., 2012], but it has its own short-comings (see §2.4.2.1). It is possible in ILNP as it provides mobility as a first class service [Phoomikiattisak, 2016]. The differences in semantics between IP addresses and ILNP namespaces enable our solutions to independently deal with individual clients and a server’s topological connectivity. Our solutions tie a client’s session to an ephemeral destination identifier of the server and/or to ephemeral locator of the server at different intervals either using DNS at the start of the session or through ILNP-specific locator update messages during the session.

In our research, we will take DNS fast flux and ILNP namespaces to achieve DoS mitigation. Each ILNP *capability*, within each defence, will switch frequently in a short period while maintaining MTD objectives such as, connection continuity. The frequent switching of capabilities is one of the MTD requirements (see §2.4.1.1).

1.7 Thesis Outline

1.7.1 Research Motivation and Approach

ILNP’s architecture allows its namespaces to be flexible in their use in the network stack and it allows an MTD-based DoS defence solutions to be orchestrated. ILNP’s first-class mobility and connection-continuity allows us to use ILNP namespaces as DNS *capabilities* and DNS fast flux in a way which has not been used before. We used this motivation and approach to create defence solutions that will be empirically evaluated in Chapters 3 to 5.

1.7.2 Research Questions

Based on our rationale and motivation, we formed the following research questions.

1. How can the approaches of MTD, DNS fast flux, and the concept of *capabilities* be used with ILNP namespaces to mitigate low-bandwidth SYN floods and bandwidth-exhaustive volumetric attacks?
2. Do ILNP’s identifier-based DNS *capabilities* provide a defence solution against SYN flood attacks that can be used as an alternative to the existing SYN Cookies-based defence?
3. Do ILNP’s locator-based DNS *capabilities* provide a defence solution against volumetric DoS attacks which is not possible with the first-class features of IPv6?
4. Is it possible to combine ILNP’s identifier-based capabilities and locator-based capabilities to simultaneously mitigate low-rate SYN floods and bandwidth-exhaustive volumetric attacks on an enterprise host and its network?

Although SYN Cookies can be used with both IPv6 and ILNP, there is a possibility of eliminating SYN Cookies through native features of ILNP. There are drawbacks (see §3.1) associated with SYN Cookies, whereas ILNP provides an opportunity to eliminate these drawbacks (since it does not modify TCP SYN-ACK packets used in the connection establishment).

We will investigate ILNP’s locator-based DNS capabilities which might not be possible with the first class features of IPv6. Our discussion sees these defences in the scope of IPv6 but our defences are entirely based on ILNP. We chose IPv6 because of its increased adoption in real networks and due to the fact that engineering of ILNP can also be done through IPv6.

1.8 Thesis Structure

Chapter 1 covered relevant concepts of this research including security challenges faced by enterprises (i.e., host and network security in general) and importance of DoS defence through the use of ILNP and DNS. It is also accompanied by motivation, research questions, and thesis structure.

Chapter 2 covers background knowledge on DoS, ILNP, and DNS. It provides state of the art in enterprise DoS defence, classifies some common attacks and provides advantages and disadvantages of their mitigations. The chapter 2 also consolidates the importance of MTD approaches and emphasises on simplicity of using already established DNS infrastructure for MTD approaches.

Chapter 3 shows how ILNP-based node identifiers can be used as ephemeral DNS capabilities to mitigate SYN flood attacks. It shows the conceptual mechanism for defence, the new network elements used, and their interactions with existing systems. It also shows a real world implementation of the approach and technologies used behind them. It provides information on what security benefits we can achieve using identifier-based approach. The chapter explains the experiment design, the testing, the empirical results obtained, and the chapter summary. The research question 2 is addressed in this chapter.

Chapter 4 shows how ILNP-based topologically significant node locator values can be used as ephemeral DNS capabilities to mitigate traffic based volumetric UDP and SYN flood attacks. The chapter shows what benefits we can achieve by using only locator values. Similar to chapter 3, it provides the information about the entities involved, their interactions, the experiment design, the testing, the empirical results and the chapter summary. The research question 3 is addressed in this chapter.

Chapter 5 presents how ILNP-based identifier and locator values can be used together to provide multi-layered security. It also shows the conceptual mechanism, the network elements involved, their interactions, the experiment design, the testing, the empirical results obtained, and the chapter summary. The research question 4 will be addressed in this chapter.

Chapter 6 provides the conclusion, the list of contributions made in this work, and the discussion of defences' specifications chapters showcased in this dissertation. It then moves on to future works that this research can entail.

Appendix A provides implementation details of extra network message spec-

ifications used in ILNP's identifier-based capabilities.

The research question 1 will be a common theme in all chapters.

Chapter 2

ILNP, DNS, And DoS Attacks

This chapter describes the ILNP protocol along with its specific functionalities which can be used to defend against DoS attacks. It then describes the DNS infrastructure, its relationship with ILNP, and its usability to defend against the DoS attacks.

The state of the art of the DoS attacks along with its classification, challenges, and defences will also be described. The security paradigms for effective DoS mitigations will also be covered, along with their viability when used in the context of ILNP and DNS.

We show the problem space we address with our contributions within the current systems landscape by a critical analysis of the current approaches to DoS defence.

2.1 Identifier-Locator Network Protocol (ILNP)

When this research was started in October 2014, IPv6 was deployed in $\sim 4.65\%$ ¹ of the Internet. As of this writing, $\sim 25\%$ of the Internet has been deployed with IPv6 i.e. it is now capable to handle IPv6 traffic. Apart from its adoption, there has been an increased interest in the development of DoS defences which are based on IPv6. Due to these reasons, our research will only consider ILNP which is a superset of IPv6. It will provide qualitative comparison among these protocols where necessary.

IPv6 resolved the problem of address depletion in Internet Protocol version 4 (IPv4), and provided *extensions*-based solutions to mobility, e.g., MIPv6 [Johnson et al., 2004], multihoming [Nagami et al., 2007], and security, e.g., Internet Protocol Security (IPSec) [Kent & Atkinson, 1998]. The IPv6 extensions have complex architectures and implementations. Apart

¹<https://www.google.com/intl/en/ipv6/statistics.html>

from protocol extensions, lack of support for inter-domain traffic engineering, scalability, multihoming, and TCP connection continuity are some of the most important challenges faced by IPv6. We explore recommended [Li, 2011] ILNP architecture whose specific advantages in these areas are being critically researched. Some of its promising advantages include:

- First class (native to the protocol and not extension-based features) and scalable host/network mobility [Phoomikiattisak, 2016].
- Decoupling of host/network identity and location [Atkinson & Bhatti, 2012a].
- Traffic engineering through the use of naming [Atkinson et al., 2009b].
- Harmonized integration of multihoming and mobility [Atkinson et al., 2009a].

In this section, we provide a comprehensive description of ILNP concepts, and its operations along with its specific well-researched and peer reviewed mechanisms which we employed in our research.

It should be noted that the ILNP has two engineering instances, ILNPv4 for IPv4 inter-operability, and ILNPv6 for IPv6 inter-operability. In terms of architecture, we will refer to it as ILNP and we will use the version-specific suffix where necessary.

2.1.1 ILNP Architecture

ILNP provides two distinct namespaces which eliminate the need for an IP address. These namespaces are used to name different objects within the Internet. An object might be a physical or virtual node. A node is an entity that can be named, e.g., a (physical or virtual) server or a (physical or virtual) network. It also supports composition of multiple physical nodes into a virtual object.

2.1.1.1 Node Identifier (NID) Namespace

A Node Identifier is a set of bits that uniquely identify a node but not its network topological location. The NID values are used at the transport layer of the network stack, and they are invariant during topological changes in connectivity of the node within the same ILNP session. An ILNP session is a session where both end hosts use the ILNP protocol. NID values are not visible below the transport layer, so they are not used for routing network packets. Similarly, they cannot be used to name the (physical or virtual) network interface.

A node can have more than one NID value that can be active simultaneously, with each NID value used for a unique ILNP session. A node

that starts an ILNP communication session can learn the NID values of the correspondent node through DNS or any other name resolution mechanism. An ILNP-enabled respondent node can learn the source NID value of the node that starts an ILNP session from the received ILNP packet.

NID values are not strongly bound to network interfaces. Instead, they can have dynamic bindings with one or more interfaces in the same ILNP session. This dynamic binding is used to ensure TCP connection continuity during changes in a node's network connectivity, as ILNP mandates the use of NID values at the Transport layer only.

2.1.1.2 Locator (L) Namespace

A Locator (L) value is a set of bits that can be used for packet routing, hence they are topologically significant names. They are similar to IP network routing prefixes supporting longest-prefix match routing. Locator values are only used at the Network layer, but they are not visible above it.

Locator values can change within the same ILNP session, i.e., a single NID value can have more than one active L values. When a node can change its L value(s) within the same ILNP session, it enables harmonized mobility and multihoming (cf. §2.1.3).

A communicating-node, whether client or server, informs the correspondent node about changes in its network connectivity through a control message called Locator Update (LU) within the ILNP session. An LU message contains new network information, e.g., one or more Locator values, and enables the respondent ILNP nodes to modify their dynamic bindings. Once corresponding dynamic bindings are updated, the respondent node can send an LU ACK (Acknowledgement) message back to the correspondent node.

2.1.1.3 Solving Address Entanglement In IP Network Stack

The use of namespaces and addressing information in the ILNP and the Internet Protocol (IP) is shown in Table 2.1. In the IP architecture, an IP address is used at the transport, the network, and the physical layers. A change in the IP address, in any layer, will disrupt the IP session unless some IP extension is used to manage this disruption. In IP, the application layer can use IP addresses and Fully Qualified Domain Names (FQDNs) in software configurations. On the contrary, ILNP mandates that the NID and L values should only be used at the transport and network layers, respectively. It is recommended that FQDNs should only be used at the application layer. The NID values are dynamically bound to the L values (in many-to-many fashion) while the physical layer must use L64 to network interface dynamic bindings. This allows a crisp separation of concerns in the network stack.

Protocol Layer	ILNP	IP (v4 & v6)
Application	FQDN	FQDN, IP address
Transport	Identifier, NID	IP address
Network	Locator	IP address
(Interface)	dynamic binding	IP address

Table 2.1: Use of namespaces and addressing information in ILNP & IP [Atkinson & Bhatti, 2012a]

2.1.2 ILNPv6 — An Engineering Instantiation of ILNP

ILNPv6 is a superset of IPv6 [Atkinson & Bhatti, 2012a] because it has native support for all the native functionalities of IPv6 and important IPv6 extensions, e.g., mobility and multihoming. It also provides features not possible with native IPv6 such as, inter-domain scalability of the Internet core network [Li, 2011]. It is also inter-operable with the IPv6 infrastructure, hence incrementally deployable. In addition, it also solves the issue of TCPv6 connection continuity during changes in the network connectivity.

ILNPv6 namespaces are derived from an IPv6 128-bit address by splitting it in half. The upper 64-bits of the IPv6 address are used to create an ILNP node locator namespace (L64), while the lower 64-bits of the same address are used to create an ILNP node identifier namespace (NID64). Figure 2.1 shows encodings of ILNPv6 namespaces and IPv6 address.

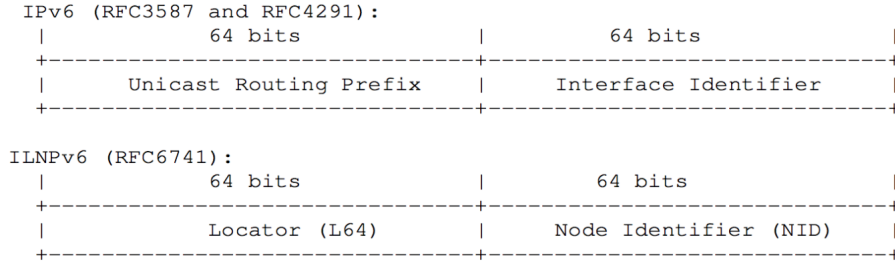


Figure 2.1: IPv6 address space and ILNPv6 namespaces [Atkinson & Bhatti, 2012b]

L64 values have the same syntax and semantics as IPv6 unicast routing prefixes, whereas NID64 values have the same syntax as IPv6 Interface Identifiers but different semantics. ILNPv6 namespaces are encoded into the same 128-bits of an IPv6 address. To identify an ILNPv6 packet, ILNPv6 specifies a 96-bit *nonce* value in the optional IPv6 destination header. Locator Update and respective Acknowledgement control messages, which enable harmonized ILNP mobility and multihoming, are encoded in the In-

ternet Control Message Protocol version 6 (ICMPv6). ILNP permits the use of any other transport mechanism for locator updates. Our research recommends the use of a secure transport mechanism for locator updates.

Apart from the addition of a nonce value in the optional IPv6 destination headers, there are no modifications in IPv6 headers, so the network will perceive an ILNPv6 packet as an IPv6 packet. This approach enables incremental deployment of ILNPv6 in the current Internet, and makes ILNPv6 a host based protocol.

For deployment purposes, ILNPv6 is implemented by modifying the IPv6 codebase. ILNPv6 prototypes are available for Linux [Phoomikiattisak, 2016] and FreeBSD [Simpson, 2016], and our research uses the modified Linux kernel version 4.9.38, which was ported by myself from the modified Linux kernel version 3.9.0 implementation developed by Dr. D. Phoomikiattisak.

2.1.2.1 Identifier-Locator Communication Cache (ILCC)

Each ILNP node maintains the required information for an ILNP operation in the Identifier-Locator Communication Cache (ILCC). In our ILNPv6 kernel implementation, it is implemented as a linked list data structure containing the following information:

- NID64 and L64 values of the node, and their dynamic bindings.
- Dynamic bindings between namespaces of source and destination nodes for each ILNP session.
- Dynamic bindings between L64 values and network interfaces.

In current ILNPv6 kernel variants, these bindings are modified based on router advertisements, Locator Update messages, domain name resolution responses, and ILNPv6 connection requests. In the future, the bindings in the ILCC can be modified based on the traffic engineering requirements of new host/network functionalities.

2.1.3 ILNP Mobility

ILNP natively supports node mobility through a crisp separation of NID64 and L64 values in the network stack, where a node can be a (physical or virtual) host or a (physical or virtual) network. ILNP also supports session continuity in a mobile node within another mobile node, e.g., a mobile device within a mobile vehicular network.

ILNP mobility requires a node *handoff* mechanism and dynamic DNS updates of the L and I records (see §2.2.3). A handoff mechanism mandates any kind of operation through which existing sessions are shifted to a new network path. ILNP enables handoff using Locator Updates.

ILNP enables mobility through dynamic updates in an ILCC cache via Locator Update control messages, and in client/server environments through dynamic DNS Resource Record (RR) updates [Atkinson et al., 2012] (also see §2.2.3).

Our research mitigates DoS attacks with the help of ILNP mobility by using the handoff mechanism and dynamic DNS updates of the L and I values.

2.1.3.1 Host Mobility

Host mobility occurs when an individual host changes its network connectivity (i.e., the point of attachment), which can be done through the handoff mechanism. There are two types of handoffs, hard handoff and soft handoff. In a hard handoff, a host completely breaks its connectivity with the currently attached network before attaching itself to a different network. In a soft handoff, a host maintains connectivity with the current network until it attaches itself to a different network. A soft handoff entails minimum packet loss [Phoomikiattisak, 2016]. ILNP supports both types of handoffs.

If a host performing handoff between two same underlying technologies (e.g. WIFI) performs a *horizontal handoff*. If the underlying technologies are different (e.g., WIFI and 3G/4G) then it performs a *vertical handoff*. ILNP supports both kinds of handoffs.

Figure 2.2 shows ILNP host mobility with handoff.

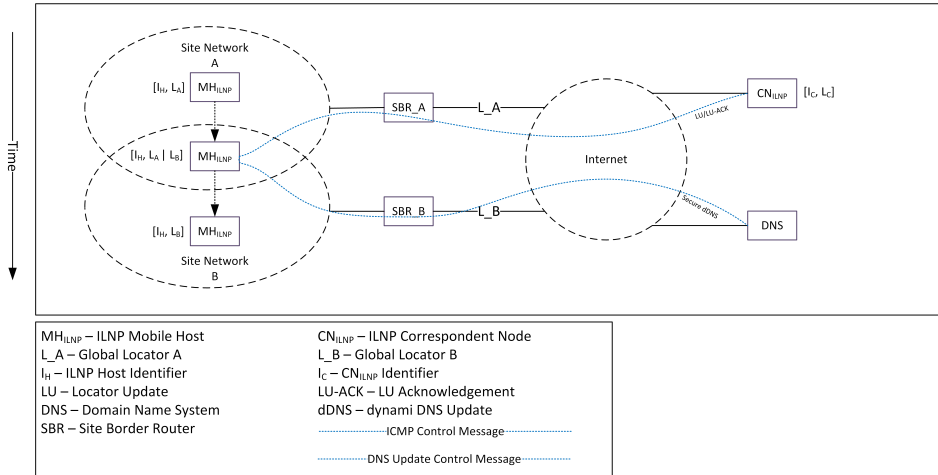


Figure 2.2: ILNP based host mobility using handoff

The mobile host MH_{ILNP} is first connected to the site network A. It is using I_H as an identifier and L_A as a locator. L_A is provided by the site network A. Once it moves to an overlapping region where it has an access to both network A and network B, then it updates its ILCC entries to

reflect new reachability information, i.e., its I_H identifier is now dynamically bound to L_A and L_B locators. Now it can use any locator value to reach correspondent nodes. Once it moves to network B , it updates its bindings by removing the dynamic bindings between I_H and L_A .

In the network overlap region of Figure 2.2, MH_{ILNP} sends locator update messages to the correspondent nodes so that all communications can use new and/or old network information. MH_{ILNP} also sends a dynamic DNS update message with new network information so that new incoming connections can reach it using either information. Once it has moved to network B , all communications take place using the L_B locator.

2.1.3.2 Site Mobility

ILNP supports site mobility where an entire site may be mobile. A site can be in hard or soft handoff mode which can itself be a horizontal or vertical handoff. Figure 2.3 shows ILNP based site mobility using handoff.

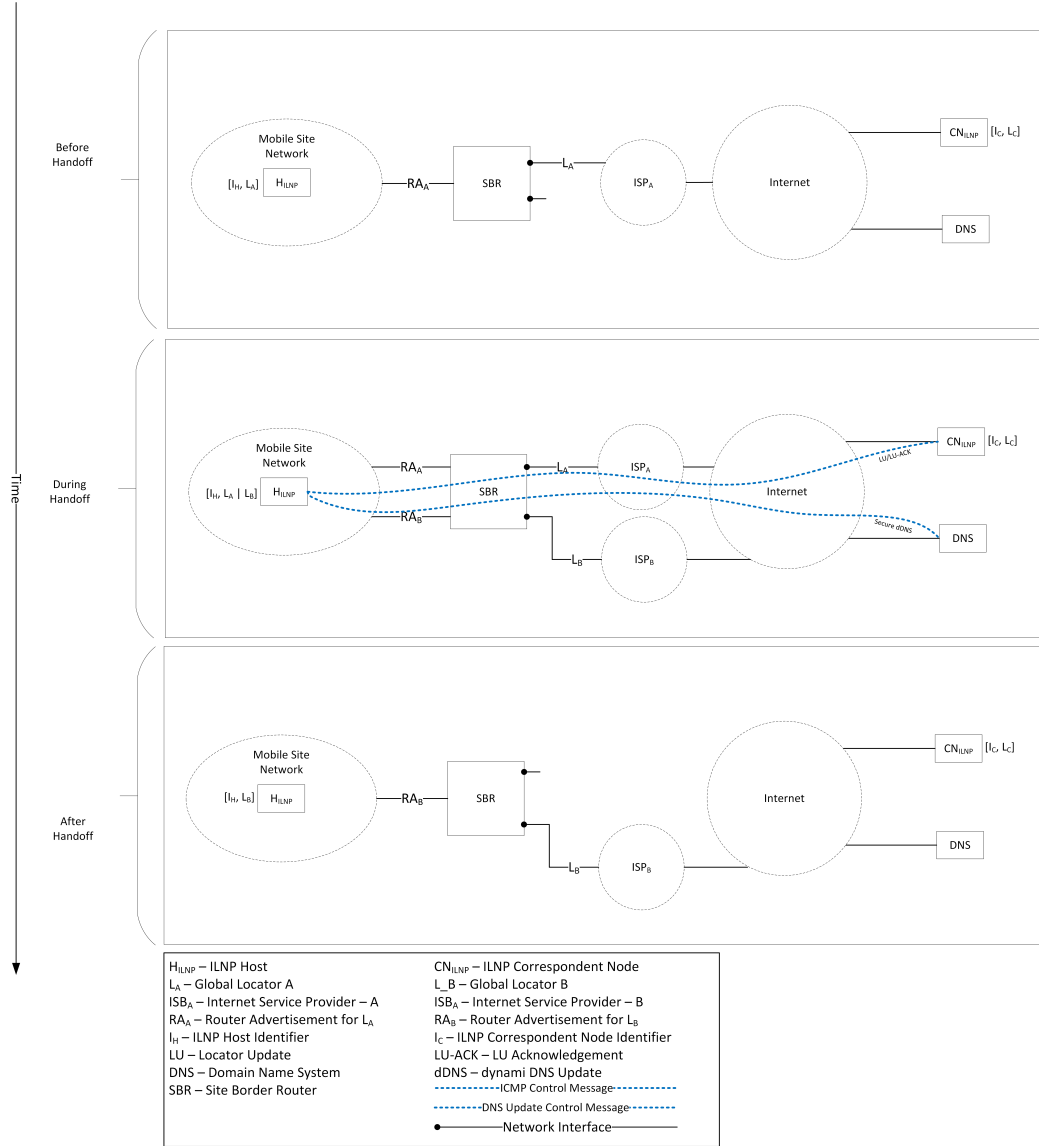


Figure 2.3: ILNP based network mobility using handoff

Before handoff, the mobile site is connected to ISP_A through the Site Border Router (SBR). The SBR is using the L_A locator provided by ISP_A . It advertises this locator to the network devices. So, each host in the mobile network is communicating with correspondent nodes using ISP_A .

Once the mobile site moves to an overlap region where it is connected to both ISP_A and ISP_B , each host within the mobile network has the corresponding dynamic bindings, i.e., their identifiers are dynamically bound to both upstream providers (ISPs) through L_A and L_B . If the device uses both locators at the same time, then it is in soft handoff mode. If it is only

using the new one then it is in the hard handoff mode. After the handoff, all communication happens solely using L_B through ISP_B .

During a handoff, nodes within the mobile network send locator updates to correspondent nodes with existing connections. If these nodes are publicly accessible servers, then they send corresponding dynamic DNS updates for incoming connections. Each host within the mobile network behaves as if it is also mobile even though it might actually be fixed.

Our work makes use of site mobility to provide site network defence against DoS attacks that target whole site availability (see chapters 4 and 5). Our work does not prefer any type of handover over another. But we employed soft handoff to minimize packet loss while our defences are in place and to check control message packet loss in congested links.

2.1.4 ILNP And Security

Current ILNP specifications address end-to-end support for IP security during mobility, nonce-based authentication of ILNP packets, and forged Identifier attacks. These specifications do not address the use of namespaces as a DoS defence mechanism. This research fills this gap by using ILNP namespaces, and ILNP mobility in a novel way (see chapters 3 to 5).

Current ILNP specifications address the following solutions for end-to-end security. We also present a side-by-side comparison with our research:

- IPsec is supported in ILNP by removing the strong bindings between IP addresses and IPsec Security Associations (IPsec SAs). ILNP mandates the use of Identifiers in these associations rather than IP addresses in current IP security operations. IPsec SAs for IPv6 break when a node moves from one network to another. IP security for ILNP allows a node to maintain its security associations during changes in connectivity. Our research does not make use of IPsec to provide DoS defence, but recommends the use of IPsec in enterprise networks and their communications with the Internet.
- ILNP mandates the authentication of ICMP messages which enable mobility. This is achieved through the introduction of nonce value within control messages. Our research recommends to use encryption of such messages as well.
- ILNP provides support for defence against off-path attacks to ILNP sessions by having a unique nonce value in ILNP packets, and by having the same unique nonce value in ICMP control messages, as mentioned above. Our research does not break this principle, i.e., it makes use of mobility without modification to nonce values associated with an ILNP session.

- ILNP recommends the use of secure DNS dynamic update [Wellington, 2000] for nodes which are listening for new ILNP sessions (also see §2.2.3). Our research introduces a new backend processor which controls updates to the DNS. For further security of the backend processor, we make use of a TLS-enabled secure channel between DNS and the backend processor to carry compact messages that contain new DNS Resource Records (RRs). These messages are used by API interfaces available in one of the market-leading DNS software. This is only possible in enterprise environments which make use of their public-facing DNS (within a Demilitarized Zone (DMZ)) by allowing DNS to connect to its backend processor. Our research also recommends the use of secure DNS dynamic updates by the backend processor.

We believe that ILNP mobility can be utilized to enable DoS defence by fast transitioning of the ILNP sessions to secure and topologically-significant attack-free paths while an attack is happening on a compromised path. Similarly, the use of Identifiers per client and its authorization over nonce-based packet authentication can enable defence against particular types of DoS attacks along with off-path attacks.

2.2 Domain Name System (DNS)

Domain Name System (DNS) is a distributed network which provides name resolution services for the Internet. Internet nodes have numerically formatted addresses through which they can be identified and located. DNS makes their conversion between human-readable domain names and numeric counterparts possible [Mockapetris, 1983], [Mockapetris, 1987].

2.2.1 DNS Architecture

The DNS architecture is a scalable and highly available architecture that has a hierarchical and distributed global database containing Resource Records (RRs). These RRs can be IP addresses, application service names, security data points, and hostnames, i.e., Fully Qualified Domain Names (FQDNs). These RRs are contained within zone files in each DNS server (termed as DNS nameserver). Two important types of nodes in DNS are the *root* nameservers and the *resolvers* (domain nameservers).

A FQDN contains hierarchical information which is used by each layer in the DNS hierarchy. For example, *cs.st-andrews.ac.uk.* contains complete information as stored in the DNS tree/hierarchy. For the representation of a root name, a *dot* symbol is used at the end of the domain name which completes the FQDN syntax. Two examples along with supplementary in-

formation on the FQDN structure in the context of DNS hierarchy is shown in Figure 2.4.

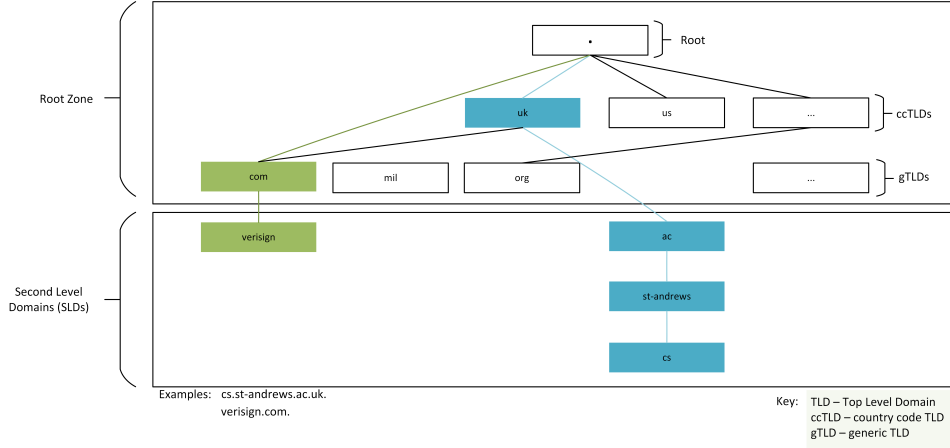


Figure 2.4: Fully Qualified Domain Names (FQDNs) in the context of DNS hierarchy

2.2.2 DNS Engineering

There are 13² root nameservers which store, cache, and provide resource records of Top Level Domain (TLD) nameservers. There are multiple types of TLDs, e.g., country code TLD (ccTLD) and generic TLD (gTLD); for example *.uk* and *.mil* are examples of ccTLD and gTLD, respectively. Each root and TLD nameserver is globally replicated in different countries for high availability. Each nameserver regularly synchronizes its local copies with the distributed infrastructure.

ISPs, government institutions, and enterprise networks contain DNS resolver nodes which resolve domain names to corresponding resource records on behalf of clients. Similarly, there are local software agents which act as resolvers as well (stub-resolvers) within web browsers or operating systems. Resolvers can be implemented as *recursive* or *iterative* (see §2.2.2.1). The responsibility of stub-resolvers is to check system files for resource records either cached or stored (e.g. in */etc/hosts* file in Linux), or to query ISP-operated DNS resolvers. An example implementation of a stub-resolver is available in *glibc*³, which is part of most Linux operating systems. In our research, we modified *glibc* to work with ILNP-specific RRs within DNS.

Nameservers maintain two kinds of information. The first kind contains authoritative information about a domain name, and the second kind contains cached resource records from previous DNS database synchronization

²<https://www.iana.org/domains/root/servers>

³<https://www.gnu.org/software/libc/>

events. Node-specific resolvers also cache resource records based on a special value in a DNS protocol message data structure called Time To Live (TTL). We would take the opportunity to emphasise that, for our research, we recommend TTL values (start of the TTL counter) to be one second or less in the resolvers, because our DoS defences rely on regular updates to resource records in nameservers.

2.2.2.1 Resolver Implementations

Recursive and iterative implementations of resolvers use different steps for name resolution. Figures 2.5 and 2.6 show both implementations. In a recursive implementation, a stub-resolver delegates the end-to-end name resolution to the ISP's nameserver, whereas, in an iterative implementation, the stub-resolver takes responsibility of end-to-end name resolution.

We take the opportunity to show the placement of our backend processor in both implementations using the following figures. It can be noted that placement of the backend processor is independent of resolver implementation. However, we recommend the use of an iterative approach because it provides more client information to the backend processor.

Recursive DNS:

Let us take an example of DNS name resolution for the *verisign.com* domain. A client enters *verisign.com* in the browser, which contacts the stub-resolver within the client operating system. Then, the following ordered steps are performed by an end-to-end recursive resolution process.

1. The stub-resolver makes a DNS client query for *verisign.com* to an ISP nameserver whose address information is stored in its configuration.
2. The ISP resolver/nameserver makes a DNS query to the root nameserver for the *.com* TLD domain (if ISP is not aware of such information).
3. The root nameserver sends a response to the ISP nameserver with address information of the *.com* TLD nameserver.
4. The ISP nameserver sends a DNS request for *verisign.com* to the TLD nameserver.
5. The TLD nameserver returns the address information of the nameserver responsible for having authoritative information about *verisign.com*.
6. The ISP nameserver sends a DNS request to the authoritative nameserver.

7. The authoritative nameserver sends address information about *verisign.com* server to the ISP nameserver.
8. The ISP nameserver sends a DNS response containing *verisign.com* server address information back to the client.

After these steps have completed, a client can initiate a connection to the *verisign.com* server by using the address information from the DNS response. Once the connection is successful, the *verisign.com* website is loaded within the browser.

Our solutions can be impacted if the ISPs employ caching of records. We recommend that the TTL values should be small enough so as to discourage caching. The low TTL values will help end-users to have an updated identity and network access information about the servers.

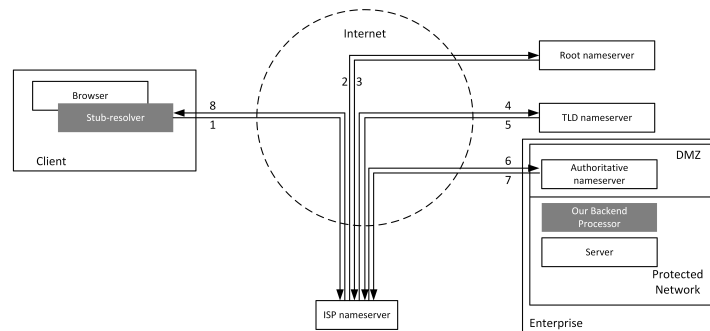


Figure 2.5: End-to-end Recursive DNS name resolution

Iterative DNS:

In an iterative DNS, every step that was performed by the ISP nameserver is performed by the client resolver. An ISP nameserver provides address information about the root nameserver to the client resolver. The following ordered steps are performed after a client enters *verisign.com* in the browser bar:

1. The client resolver makes a DNS query for *verisign.com* to an ISP nameserver whose address information is stored in its configuration.
2. The ISP nameserver sends a DNS response containing address information of the root nameserver.
3. The client resolver sends a DNS query to the root nameserver for the *.com* domain resolution.
4. The root nameserver sends a response to the client resolver with address information of the *.com* TLD nameserver.

5. The client resolver sends a DNS query for the *verisign.com* domain to the TLD nameserver.
6. The TLD nameserver returns the address information of the DNS nameserver responsible for having authoritative information about the *verisign.com*.
7. The client resolver sends a DNS query to the authoritative nameserver.
8. The authoritative nameserver sends address information about *verisign.com* server to the client resolver.

After these steps have completed, the client resolver gives this information to the browser which can initiate a connection to a *verisign.com* server by using the address information from the DNS response. Once the connection is successful, the *verisign.com* website is loaded within the browser.

In our research, our backend processor modifies the DNS response in step 8 that was sent by the authoritative DNS server.

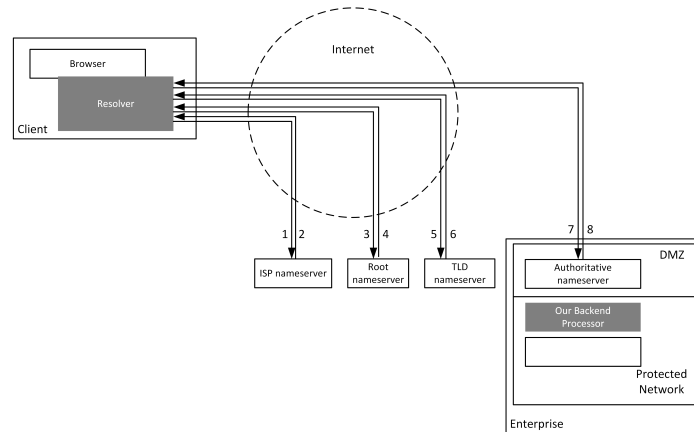


Figure 2.6: Iterative DNS name resolution

2.2.2.2 DNS Transport Protocols And Security

End-to-end DNS queries and responses can be carried out using any Transport layer protocol. The query can be an update to the DNS information in zone files, using dynamic DNS update [Vixie et al., 1997]. Our DoS defence protocols do not mandate use of any particular Transport protocol, but we recommend end-to-end security of DNS transactions.

2.2.3 DNS And ILNP

2.2.3.1 New DNS Resource Records (RRs) for ILNP

As ILNP deprecates the use of IP addresses, AAAA and A RR types are not supported. AAAA RR type is used with IPv6, whereas A RR type is used with IPv4. For nodes which expect incoming network connections, ILNP introduces the following new RRs in DNS [Atkinson et al., 2012], and they can only be used by *ILNP-aware* nodes (nodes that implement and use ILNP):

NID Resource Records

A NID is an address class independent Node Identifier (NID) RR with an RR type value of 104 in DNS packets. It is a 64-bit field with a 16-bit *Preference* field. A Preference field can be used by a domain owner for relative preference among more than one NID RRs. A domain name can have zero or more NID RRs, but there must be at least one NID RR which can be used by ILNP sessions, where each end-to-end ILNP session is only tied to a single NID. A DNS administrator can choose any TTL value for the NID RR.

Our DoS defences do not use Preference fields, and do not have requirements for specific TTL values for NIDs. Our DoS defences do require a NID field of the NID RR.

L32 Resource Records

A L32 RR is an address class independent Locator value with a RR type value of 105 in DNS packets. A L32 RR is a 32-bit Locator for ILNPv4-capable ILNP node. ILNPv4 is an engineering instantiation of ILNP which is inter-operable with IPv4. For details, please see RFC 6742 Section 2.2. Our research does not require a L32 RR, and has no requirements for its TTL and Preference values.

L64 Resource Records

A L64 RR is an address class independent ILNP node Locator value with an RR type value of 106 in DNS packets. L64 RR contains a 64-bit Locator, with a 16-bit Preference value, to be used with ILNPv6-capable nodes. Its Preference field has the same usability as a NID Preference field, and is used to prioritize L64 RRs. There is a TTL field associated with L64 RR that can be of any value chosen by the domain administrator. A domain name can have zero or more L64 RRs, but there should be at least one L64 RR active for each ILNP session, where each end-to-end ILNP session can use more than one L64 values if available.

Our DoS defences require L64 RR, but have no requirements for its Preference field, and we recommend the TTL value to be as low as zero seconds (so that the clients always get the fresh information as our defences produce reachability information after a very short interval).

LP Resource Records

A LP RR is an address class independent ILNP node Locator Pointer with an RR type value of 107 in DNS packets. A LP RR encoding is a variable length FQDN field and a 16-bit Preference field. Its FQDN is used to resolve L32/L64 RRs for a domain name, i.e., it points to a subnetwork of the ILNP node. Its Preference field has the same usability as the NID/L64/L32 Preference field, where an administrator can assign priorities to multiple LP records. A domain name can have zero or more LP RRs. DNS name compression is forbidden for LP RRs [Atkinson et al., 2012]. There is a TTL field associated with the LP RR that can be of any value chosen by the domain administrator.

Our research does not require LP RR, and has no requirement for its TTL and Preference values.

Table 2.2 shows information on presentation formats and examples of DNS resource records for IP and ILNPv6.

Term	DNS Record	Definition	Presentation Format in DNS Zone file	Example(s) (one per line)
IPv6 Address	AAAA	128-bit value to represent node location and identity in IP network	<owner-name>IN AAAA <IPv6-address>	host1.example.com. IN AAAA 2001:0DB8:1140:1000:0014:4fff:ff20:ee64
ILNPv6 Node Identifier	NID	64-bit value to represent node identity in ILNPv6 network	<owner-name>IN NID <Preference><NodeID>	host1.example.com. IN NID 10 0014:4fff:ff20:ee64
ILNPv6 Locator	L64	64-bit value to represent node location in ILNPv6 network	<owner-name>IN L64 <Preference><Locator64>	host1.example.com. IN L64 10 2001:0DB8:1140:1000 l64-subnet1.example.com. IN L64 10 2001:0DB8:1140:1000
Locator Pointer	LP	Variable length FQDN character string providing L64 indirection	<owner-name>IN LP <Preference><FQDN>	host1.example.com. IN LP 10 l64-subnet1.example.com.

Table 2.2: DNS Resource Records of IP and ILNPv6 along with their definitions, zone file presentation formats, and examples.

2.2.3.2 Modifications In The Client Side DNS APIs For ILNP

When a client makes a DNS query for a name resolution, it encodes a *question* in a DNS request packet for a specific resource record, e.g., AAAA or A. In ILNP, an ILNP node can request an ILNP-specific DNS record for another ILNP node/network. There are many client side libraries for DNS name resolution but we will cover glibc⁴ because it is supported in most Linux-based operating systems with diverse instruction-set architectures (ARM⁵, PowerPC⁶, SPARC⁷ etc).

In our research, we modified the *getaddrinfo* API to honour API calls from applications which are not ILNP-aware, i.e., do not make NID, L64, or LP queries, but require AAAA queries to enable communications. This modification allowed us to run common applications e.g. curl⁸, iperf3⁹, dig¹⁰, and ping6¹¹.

We made the following modifications for ILNP client nodes which want to establish a connection with another ILNP node through DNS name resolution:

1. When a client application makes a DNS name resolution function call for an AAAA record through *getaddrinfo*, the modified glibc intercepts the DNS response.
2. The modified glibc splits the IPv6 address from the DNS response in half, and stores the lower 64 bits as the NID, and the upper 64 bits as the L64 value.
3. The modified glibc adds the ILCC bindings (see §2.1.2.1) between the NID and L64 values of the client with the NID and L64 values of the remote node.
4. The modified glibc then returns the IPv6 address to the client application.

Once these steps have been carried out, our required legacy client applications work with the ILNP kernel implementation. The above mentioned modifications were done by myself for glibc version 2.23.

⁴<https://www.gnu.org/software/libc/>

⁵<https://developer.arm.com/products/architecture>

⁶<https://www.ibm.com/developerworks/systems/library/es-archguide-v2.html>

⁷<http://sparc.org/technical-documents/>

⁸<https://curl.haxx.se>

⁹<https://iperf.fr>

¹⁰<https://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/man.dig.html>

¹¹<https://linux.die.net/man/8/ping6>

2.3 DoS Attacks

Extortion, rivalry, rogue business competition, intention to punish, political enmity, and sometimes fun are some of the motivational factors contributing to DoS attacks. Anti-state and state-sponsored groups routinely launch offensives on critical infrastructure of other governments. They employ a DoS mechanism which is one of the attack tools used in cyber-warfare.

In order to plan a DoS attack, attackers make use of information gathered through hacking and systems vulnerability testing, e.g., port scanning. They use impersonation, compromised network machines, and physical aggression to yield successful DoS. After getting enough information to launch an attack, attackers employ either low-rate, specially-crafted data packets to exhaust the enterprise systems' resources, or high data-rate packets to congest links going towards the enterprise network.

In this research, we will propose and evaluate solutions which defend an enterprise against both low and high data-rate packets to protect both the enterprise systems and its network.

It should be noted that our research does not deal with attack detection but with mitigation. Reduction in penetration or vulnerability testing done by an attacker and the provisioning of client privacy are two of the side effects of our defences. In regards to deployment of these defences, an enterprise has the option to either execute them after it has detected an attack, or it can use them any time in order to benefit from the side effects as well.

2.3.1 History of DoS Attacks

The aim of this section is to extensively cover the history of DoS attacks from their inception to present times. We have formed a time-line view by epochs, because they represent periods around which new network functionalities either enhanced or limited DoS attacks. Each section mentions that aspect with emphasis.

2.3.1.1 1974-2000

This epoch saw a wide adoption of computing systems and networks for business and personal use. Operational knowledge of computers and intentions to circumvent Cyber systems was an attractive crime due to the limited legislation for cybercrime.

In 1974, a teenage student named David Dennis launched and reported the first DoS attack at the Computer-based Education Research Laboratory (CERL) [Radware, 2017]. He executed code which forced 31 computers in CERL to be turned off. Since then, DoS attacks have increased in frequency and complexity to cripple the availability of global enterprise networks and hosts running in them.

In 1986 [Hughes, 1992], a court hearing proceeded for a case in which the defendant erased software from a plastic circuit card in a computerized saw. This incident rendered the saw inoperable. This incident was termed as damage to property.

The most devastating incident occurred in 1988, when a student Robert Morris crippled access to more than 6,000 Internet connected systems used by academia, industry, and the government [Dierks, 1993]. The damages amounted to \$150,000 in the official documents, but they were reported to be about \$97 million in the un-official popular press.

In 1991, the Joint Academic Network (JANET) systems were accessed by an unauthorized attacker. The attacker gained the role of Systems Manager using malicious activities, and changed passwords of authorized users. This crippled the access to these systems [Hughes, 1992].

In 1992, a software contractor (Richard Goulden), after having a payment dispute with a host company, entered the enterprise premises and configured systems' passwords that were only known to him. This attack was classified as Denial of Access [Computer Weekly, 1992].

In 1997, the control tower of the Federal Aviation Administration (FAA), the emergency services of the Worcester Airport, and the Rutland community were denied services for approximately six hours. It was done by a teenage computer hacker named 'Jester'. It compromised the 'loop carrier system' (that integrated voice and data communications) of NYNEX (New England telephone company) [Conklin et al., 2018].

On February 7, 2000, Mafiaboy [Radware, 2003] made use of a distributed DoS attack to cripple the availability of Fifa.com, CNN.com, Dell, Amazon.com, eBay, and Yahoo. In the same year, Mafiaboy also launched attacks on nine of the 13 DNS root nameservers. This series of attacks were termed as NET JAM. Mafiaboy used Tribe Flood Network (TFN2) [Carnegie Mellon University, 1999] tools to launch the attack. TFN2 is a distributed set of tools that is used to launch coordinated and volumetric SYN flood, UDP flood, and ICMP flood.

2.3.1.2 2001-2003

Rootkits, worms, and DNS reflection [Al-Dalky et al., 2018] based (D)DoS attacks became more common during this period. Distributed Reflection-based Denial of Service (DRDoS), Code Red [eEye Digital Security, 2001], and Mydoom [Wong et al., 2004] collectively used UDP, TCP, and the Border Gateway Protocol (BGP) protocols through globally connected infected machines to successfully cripple a victim's availability. Rootkits provided unauthorized access, thereby enabling an attacker to execute software programs such as, worms, which endemically penetrated into other network machine. Once infected, attackers used the machines to launch attacks. A DNS reflection-based mechanisms go even further where an impersonation of

a victim's source addresses is used to launch an attack. Such an assault uses spoofing, whereby an attacker uses the fake source address (or the victim's source addresses in the case of DNS reflection) to send data packets.

Microsoft web pages, Register.com, the White House, DNS Root Servers, and the Irish Department of Finance were some of the victims in this period. In this research, we will address attacks which make use of spoofing.

2.3.1.3 2004-2008

In 2004, a junk mail worm named Mydoom infected more than one million computers, and used these infected machines to launch a Distributed DoS (DDoS) offensive.

In May 2006, a DDoS hit an anti-spam firm named Blue Security which re-directed its attacked-location to Six Apart Ltd. This in turn crippled access (around 8 hours of downtime) to Six Apart's ~1.8 million blog sites including 10 million plus registered users of LiveJournal, Typepad, and MoveableType¹².

In 2007, the cyber-warfare was started by a DDoS incident on Estonian public and private sites when Estonia disagreed with Russia on a World War II memorial relocation. Ping floods and botnet-assisted DDoS attacks were used in this cyber attack.

The SCO Group, Microsoft web pages, online game servers, DNS root nameservers, the Church of Scientology, Six Apart Limited, and institutional sites of Georgia, South Korea, and Estonia were victims of this period. In this research, we will address DDoS mitigation in Chapter 4, but we will not provide solutions for mitigation of malicious software distribution.

2.3.1.4 2009-2013

Cloud computing became mainstream after 2005, and small-to-medium enterprises started hosting their services on public, private, or hybrid Cloud architectures. In retrospect, the industry saw a rise in DoS attacks due to financially cheap computing resources enabled by cloud technologies. Since many organizations started to move their workloads to public clouds, cloud providers themselves started to see massive DDoS attacks. The attacks got more intensive in part due to an increase in Internet bandwidth as well. In 2010, Arbor Networks reported a ~49 Gbps traffic based attack on a single enterprise victim [Arbor Networks, 2010]. In 2013, Cloudflare reported [Cloudflare, 2013] that a particular Tier 1 operator saw a ~300 Gbps volumetric attack targeted towards spamhaus.org.

Due to open source development and the mass provisioning of attack tools, it became easy for novice attackers to get hold of such malicious software. Code hosting sites with collaborative coding such as, GitHub,

¹²<https://movabletype.com/>

made it possible to help develop the type of software which can be potentially exploited to do rogue activities by novice to expert offenders.

There was an increase in attacks on the application layer. Low data-rate network traffic with carefully crafted HTTP packets were used to knock out services running on application and database servers. Arbor Networks reported usage of the Hyper Text Transport Protocol (HTTP), DNS, Simple Mail Transfer Protocol (SMTP), Session Initiation Protocol (SIP)/Voice over IP (VoIP), and HTTP Secure (HTTPS) protocol based DoS attacks between 2010 and 2011.

Cloud computing provider GoGrid, DNS hosting company UltraDNS, domain registrar Register.com, web hosting firm The Planet, The Pirate Bay (pirated content torrent indexer), Iran's government websites, the U.S.A. Department of Transportation, the U.S.A. Federal Trade Commission, The Washington Post, the New York Stock Exchange, Twitter, Google, and Facebook were some of the victims in this period.

The main concerns for the lack of security controls were attributed to the lack of skilled security experts in building and maintaining effective solutions. The use of the application layer to execute DDoS attacks and the discovery of new system vulnerabilities increased the gap between having a skilled workforce to thwart these attacks and protecting the enterprise.

Most of the attacks used IPv4 as the underlying protocol whose design did not account for the DDoS challenges that came later. DDoS attackers manipulate packet headers for which there are no security mechanisms available for authentication. Most of the enterprise security controls were based on firewalls, gateways, and ISP-managed security appliances which did not incorporate attacks which used advanced techniques at the Transport and Application layers. Unfortunately, it is still the case.

We took these considerations into account and designed our solutions by selecting ILNP as the underlying protocol that has a mechanism to control traffic at the network level. We also chose DNS capabilities and Moving Target Defence (MTD) as a paradigm through which we can achieve protocol level authentication of packets and live migration of services to topologically accessible links.

2.3.1.5 2014-2018

In 2014, independent media enterprises in Hong Kong saw a ~500 Gbps DDoS attack. On Christmas day of 2014, PlayStation and Xbox Live services were rendered offline for several days by a DDoS assault. It was termed as a marketing gig from hackers to capture new DDoS customers.

In 2015, 19,000 French websites were attacked for a week in the wake of the Charlie Hebdo controversy. In the same year, Anonymous conducted a ~40 Gbps Root DNS attack on Turkish servers [Karat, 2016].

Arbor networks recorded ~124,000 DDoS attacks in 2016. In the same

year, Dyn DNS was also attacked using the Mirai botnet [Dyn, 2016] (with $\sim 100,000$ infected IoT devices). This attack affected sites like GitHub, Etsy, Twitter, and Spotify.

Peak attack traffic was intensified in comparison to previous years. In 2016, the information security blog named Brian Krebs¹³ was hit by a ~ 620 Gbps traffic based attack [Krebs, 2016].

In 2017, Brazil saw a peak of a 641 Gbps assault. In 2018, GitHub was rendered unavailable in a ~ 1.3 Tbps DDoS attack [The Hacker News, 2018]. Later that year, Arbor networks reported an intensity of ~ 1.7 Tbps on a major U.S. service provider.

Russian Banks, Turkish sites, security service providers, Rio Olympics' sites, Clinton and Trump campaign sites, the Brian Krebs site, DynDNS, DreamHost, Melbourne IT, the U.K. National Lottery, and Boston Globe were some of the major victims.

The majority of the attacks launched during this period were UDP floods, SYN floods, ICMP floods, and DNS reflection.

Table 2.3 summarises the DoS history presented in this section.

¹³<https://krebsonsecurity.com/>

Year Range	Type of Attacks	Affects and/or Affected Parties
1974-2000	Remote Server Lockdown (CERL, 1974), Software erasure from plastic circuit card (Cox v. Riley, 1986), System crash (Internet crash – Robert Morris (Attacker), 1988), Remote Control via SYSMAN impersonation (JANET – R. v. Whitely, 1991) Password Reset (R. v. Goulden, 1992), Series of remote commands to control 'loop carrier system' (NYNEX, Worcester Airport, and Jester, 1997), Distributed DoS – SYN, UDP, and ICMP floods (NET JAM, 2000)	31 servers (CERL), Inoperable computerized saw (Cox v. Riley), Over 6,000 Internet machines (Robert Morris), Main control workstation (R. v. Goulden), Unauthorized admin access to approx. all systems (JANET), 6 hour disruption (Worcester Airport and Jester), Several hours disruption (NET JAM)
2001-2003	Distributed Reflection-based DoS, Code Red (2001), UDP floods, TCP SYN floods, BGP protocol based attacks	Downtime for Microsoft, Register.com, White House (U.S.A.), DNS Root Servers, Irish Department of Finance
2004-2008	Mydoom infected 1 million machines – DDoS (2004), DDoS using packet floods (Blue Security and Six Apart Ltd., 2006), First DDoS based Cyber-warfare attack with ping floods (Estonia, 2007)	Approx. 8 hours downtime (Blue Security and Six Apart Ltd.), The SCO Group, Microsoft, online video games, DNS Root Servers, Church of Scientology, Estonia government, South Korea
2009-2013	DDoS aided by Cloud Computing resources, Application Layer DoS, SMTP, SIP, VoIP, HTTPS, Volumetric DDoS with approx. 49 Gbps attack (2010), Volumetric DDoS with approx. 300 Gbps (2013)	Cloud Providers, Tier 1 Operator with approx. 300 Gbps (2013), GoGrid, UltraDNS, Register.com, The Planet, U.S.A. Department of Transportation, Twitter, Google, Facebook
2014-2018	Volumetric DDoS with approx. 500 Gbps attack (Hong Kong, 2014), Volumetric DDoS with approx. 40 Gbps (Turkey, 2015), Approx. 124,000 Volumetric DDoS attacks (2016), Mirai Botnet based attack (2016), Volumetric DDoS with approx. 620 Gbps (2016), Volumetric DDoS with approx. 641 Gbps and 1.7 Tbps(2017), UDP, SYN, ICMP floods, and DNS reflection based attacks	Media enterprises in Hong Kong (2014), PlayStation, Xbox Live, French websites, Turkish Servers, Dyn DNS, GitHub, Etsy, Twitter, Spotify, Brazilian websites, Russian banks, Rio Olympics' Sites, Brian Krebs site, U.K. National Lottery, Boston Globe

Table 2.3: Summary of some of the DoS attacks from year 1974 to 2018

2.3.2 Attack Vectors And Classification Of DoS Attacks

DoS *attack vectors* are a means to launch DoS attacks and their variants. A multi-vector attack employs more than one vector and comprises more than 80% of the attacks on hosts and networks [Imperva, 2015]. DoS attacks use network/host resource exhaustion by using various communication or application protocols [M. Handley, 2006] (also see §1.1). They might target either different Internet layers of enterprise hosts or legitimate client access to the victim site itself.

A DoS attack can be classified based on different attack vectors [Zargar et al., 2013]. Its classification can also be in the context of wireless infrastructure environments [Geng et al., 2002], sensor networks [Wood & Stankovic, 2004], dynamic spectrum access systems [Zargar et al., 2009], VoIP and IP Multimedia Subsystem (IMS) services [Sisalem et al., 2009], and collaborative environments (e.g. space research and military applications) [Arun & Selvakumar, 2009]. In this section we classify them in a general, which is applicable to enterprises with diverse environments.

The following are some of the attack vectors:

2.3.2.1 Low Data-Rate Transport Layer Attacks

Low data-rate DoS attacks make a victim process fake packets. During processing, the victim host might be unable to service either some or all of its legitimate clients. Some attacks target the protocol data structures within the victim kernel, e.g., an attacker might send TCP SYN messages to circumvent the server's connection establishment behaviour.

Low data-rate SYN flood targets a TCP data structure called the TCP Transmission Control Block (TCB) in the victim's kernel allocated memory. A TCP connection handshake allows a client to send the first packet containing a SYN message. In response to this, the victim acknowledges with a SYN ACK message. During this period, the victim puts this half-open connection in TCB [Eddy, 2007]. The TCB is of limited memory (usually 1300 Bytes in Linux) and there is a configurable limit to how many TCBs can be created, i.e., how many parallel connections are acceptable. Usually, in Linux, there is a concept of a backlog which can be set by the *listen()* system call in the kernel. Even though a server might have a large Random Access Memory (RAM), it would have a limited backlog set by the application which is listening for connections. An attacker is only concerned about exhausting this backlog, and once this gets exhausted, a server starts sending RSTs to any new clients. The attacker uses IP spoofing to send SYN packets, so a server's SYN ACK messages are never honoured by spoofed sources. Chapter 3 provides an evaluation of an ILNP-based defence which can protect a victim against low-rate SYN floods.

2.3.2.2 Volumetric Attacks

Volumetric attacks target the bandwidth of an enterprise access network. A flood of fake network packets is sent to a victim that render the victim unavailable to legitimate clients. These attacks are normally mounted using network or transport layer packets such as, UDP floods, ICMP floods, SYN floods, etc [Wang et al., 2015]. Chapters 4 and 5 present evaluations of defences for UDP and TCP SYN-based volumetric attacks.

2.3.2.3 Packet Fragmentation Attacks

In this type of attacks, an attacker exploits the IP datagram fragmentation mechanism. Every network imposes a limit on the datagram size called the Maximum Transmission Unit (MTU). Any datagram larger than this size will be fragmented and transmitted separately. IP fragmentation is a procedure to break larger IP packets into smaller packets. These smaller packets then get reassembled at the destination. The indication to reassemble the smaller packets is indicated by the source using reassembly flags in the packet. An attacker sends UDP or ICMP packets which are larger than the allowed MTU. These packets are unable to be reassembled because there is no tracking of reassembly flags set in the victim, so the victim's resources get exhausted. [Atlasis, 2012] covers attack possibilities on IPv6 implementations using fragmentation. [Gont, 2013] covers implications of IPv6 Neighbour Discovery attacks using IPv6 fragmentation.

Our research did not propose or evaluate any ILNP-based defences against fragmentation based attacks but it can be done as part of future research.

2.3.2.4 Application Layer Attacks [Ranjan et al., 2009]

Application layer attacks exploit application layer software vulnerabilities found in victim systems to achieve DoS. These attacks are sophisticated, hard to launch and detect. A few examples are basic HTTP floods, randomized HTTP floods, cache-bypass HTTP floods, WordPress eXtensible Markup Language - Remote Procedure Call (XMLRPC) floods [Durgekova et al., 2012], SQL search attacks, Mass Content Request attacks, etc [Yang et al., 2013].

In the majority of cases, HTTP GET/POST requests are generated so as to exhaust the application server resources. The victim processes such messages and then responds, which further exhausts the victim's resources and the attached channel bandwidth. The attacker uses compromised clients (rather than spoofing) since, in HTTP, it is a requirement to have some connection state before the execution of an HTTP request.

In this research, we do not evaluate mitigations against application layer attacks but we recommend a future investigation.

2.3.3 Vulnerability Testing And Penetration Testing

Vulnerability testing is a process to identify and quantify vulnerabilities in a system. The TCP Control Block (TCB) overflow problem is one such example of a vulnerability.

In DoS, it is important to gather knowledge about a victim that can be used to find different attack vectors for launching a successful DoS attack. To gain such knowledge, an attacker uses vulnerability testing. Vulnerability testing may include port-scanning and network probing, where port scanning finds open ports in the network, and network probing might include enlisting active namespaces/addresses in the network.

Penetration testing is a process used by enterprises themselves to gain knowledge about the vulnerabilities present in their systems.

Our research does not directly influence vulnerability testing or penetration testing. But it discourages an attacker or an enterprise from performing such actions. It is important to investigate how our approaches can work to discourage external entities from vulnerability testing while allowing enterprises to perform penetration testing without hassle.

2.3.4 DoS Detection

There might be different detection schemes for individual attack vectors. An attacker either uses compromised clients or IP spoofing for attacks. Attack diagnosis is a primary requirement for any defence. In compromised clients, an attacker first installs a piece of malware (malicious software) on the systems and then controls them to direct an attack. It should be noted that there is a difference between detecting an attacker's identity and detecting the type of an attack. In this research, we cover mitigation rather than any form of detection. We assume that an enterprise has detected either a spoofing-based low-rate SYN flood, or a spoofing-based volumetric UDP flood or SYN flood (as we are only dealing with these types of attacks).

There are many detection techniques available, e.g., [Jyothi et al., 2016], [Marnerides & Mauthe, 2016], [An & Weber, 2016], [Nezhad et al., 2016], [Cheng et al., 2016], [Liu et al., 2016] etc. We do not prefer any one type of detection over another as a pre-condition to the usability of our solutions.

2.3.5 DoS Mitigation

An enterprise might use DoS prevention proactively by eliminating victim application vulnerabilities, by having ingress filtering of known attack sources, or by deploying redundant infrastructure. It is a challenging task to have a mechanism which always fixes application vulnerabilities as they happen.

Given the increase in intensity and complexity of DoS attacks (see §2.3.1), having redundant resources is expensive. Similarly, an enterprise has to

have complete information about the attack source in order to do successful ingress filtering. Some of the proactive approaches in literature are proactive specification-based fuzzing in Next Generation Networks (NGN) [T. Rontti, 2012], route-based filtering [Park, 2003], SYN Cookies [cert.org, 1996], and test-based differentiation techniques [Shevtekar & Ansari, 2007].

An enterprise can also decide to implement a reactive strategy which requires an enterprise to have DoS detection systems in place to trigger a defence mechanism. Some reactive approaches direct traffic to security providers (e.g., Cloudflare¹⁴) which have traffic scrubbing based mitigations in place [Zilberman et al., 2017]. Other defences shift all traffic from one link to redundant links, or to black-holing all traffic.

In our research, we evaluated defences which neither detect an attack nor identify an attacker but they can be used either proactively or reactively in the face of an imminent or on-going attack.

2.3.6 Existing Mitigations Against Low Data Rate SYN Floods

2.3.6.1 SYN Cookies

Using SYN Cookies [D. J. Bernstein,], [Eddy, 2007], a victim host does not allocate any state for a new SYN packet in its kernel's TCB data structure. Instead, it encodes information required by the TCB for a half-open connection in a sequence number of a SYN-ACK. If a SYN initiator is a legitimate client, then this client will send an ACK (with a sequence number of one greater than the received sequence number in the SYN-ACK) after the victim's SYN-ACK packet. This ACK message from the client will give the necessary information to construct a complete TCB at the victim.

The sequence number is a 32 bit value. According to TCP specification, a server can send any sequence number value. SYN Cookies make use of this feature and encode certain information in the 32 bits such as, Maximum Segment Size (MSS), server's IP address and port number, client's IP address and port number, etc. If the server receives the same sequence number in the ACK then it can re-construct the same information which is required to build the TCB.

SYN Cookies are implemented in Linux and FreeBSD, as configurable options.

SYN Cookies are not compatible with TCP options. They also create problems in unidirectional data flow when Selective ACKs are in use. SYN Cookies do not support application data piggybacking on the SYN packets. We will further cover problems of SYN Cookies and the rationale to use our approach in §3.1.

¹⁴<https://www.cloudflare.com>

2.3.6.2 SYN Caches

In [Lemon, 2002], Jonathan Lemon demonstrated that a linear chain of incomplete connections can be replaced with global hashtables in operating systems instead of a per-socket data structure, i.e., TCB, hence minimizing the initial connection state. It delays complete TCB allocation until the connection is completed. Host-specific secret bits are hashed with source and application service information. This hash value is used to tally a hashtable for incomplete TCBs. The oldest entry in the hashtable is purged after a specific bucket limit is exceeded.

An attacker has to know the number of secret bits in order to overflow the hashtable. These secret bits, along with the hashtable tally-mechanism makes SYN Caches an effective technique. In Lemon's evaluation, a legitimate connection took $\sim 15\%$ longer (which contributed to high latency) during an attack as compared to absence of an attack. SYN Caches do not support data piggybacking in SYN packets which is also not supported in SYN Cookies.

Since our research covers SYN flood mitigation, §3.1 provides a rationale and evaluation for our alternative to SYN Cookies. We do not provide any comparison to SYN Caches, but our defences do allow data piggybacking in the connection establishment phase. We chose SYN Cookies instead of SYN Caches because SYN Cookies are widely adopted in systems rather than SYN Caches (mostly supported in FreeBSD). We propose a comparative analysis of our approach and SYN Caches as future work.

2.3.7 Existing Mitigations Against Volumetric DoS Attacks

Two of the common DDoS defence mechanisms are filter-based, and capabilities based [Zargar et al., 2013]. Each mechanism can be used to mitigate volumetric DDoS attacks. This section will provide a discussion about defence methods that can be from either mechanism, with or without the use of DNS.

2.3.7.1 Round Robin DNS (RRDNS)

Round-Robin DNS (RRDNS) is a traditional way of load balancing [Brisco, 1995] and load distribution [Kwan et al., 1995] but can also be used for DoS mitigation. In this mechanism, multiple IP addresses of the victim are used by DNS. A different IP address is returned in the DNS response after every new DNS request.

The DNS cycles through the list of IP addresses in a round-robin fashion. So, this technique might effectively distribute the attack traffic. A victim service runs on a distributed set of servers therefore a DoS attacker has to mount an attack on all the servers in order to entirely disrupt the service.

The round robin approach also uses low Time To Live (TTL) values so that the clients always get a randomized list of addresses.

Some of the problems associated with RRDNS are

- It does not work well with dynamic web content and state-full services since servers are required to work in synchronization to serve client sessions. A single server under attack will severely affect such synchronization.
- If any one or more servers are down, then clients will still get old DNS records unless there is a mechanism to drive DNS with new lists dynamically.
- An attacker can reach any server at any time without going through name-resolution, i.e., there is no destination-controlled client authorization.
- It requires multiple physical/virtual servers to back the victim service. Some of the servers will be busy handling clients and few of them might be idle, hence resources are expected to be wasted.

In our research, we borrowed the concept of per-session round robin allocation of IP addresses, and applied it to ILNP NID64 namespaces in chapter 3. Our solution does not suffer from the above-mentioned issues due to the use of unique properties of ILNP, DNS capabilities, and the MTD (see §2.4.1) paradigm.

2.3.7.2 Reverse Proxy Through A Cloud Provider

In this technique, attack traffic is diverted to a DoS protection provider which employs distributed scrubbing data centres for DoS mitigation, e.g., [Zilberman et al., 2017].

Scrubbing is a mechanism to separate legitimate client traffic and attack traffic through Deep Packet Inspection (DPI), Access Control List (ACL)s, client puzzles, etc. Scrubbing is done at a dedicated scrubbing centre through external security providers. Proxy agents in scrubbing centres send the legitimate traffic to the victim and block the attack traffic. It is a commercial service provided by leading cloud service providers such as, Akamai¹⁵.

Scrubbing requires a redirection of traffic to the cloud scrubbing centre which is nearest to the attack source. This redirection can be done through DNS or Border Gateway Protocol (BGP).

¹⁵<https://www.akamai.com/uk/en/products/cloud-security/ddos-protection-service.jsp>

In the BGP-based method, BGP prefix announcements are used. This technique requires the control of Autonomous System (AS) by the customer, since it is the customer who has to announce this prefix. The customer normally delegates this to the DoS protection provider. That prefix range holds the scrubbing agent's addresses.

Deployment of this defence mechanism is at the network aggregation points, e.g., a peering edge link under the supervision of the Internet Service Provider (ISP) [Fayaz et al., 2015]. Scrubbing mechanisms sometimes involve the inspection of Application layer data as well, which means that the provider has complete access to the customers' data. This is an open attack vector to the privacy of the said data. Even TLS keys might be available to the security service provider, if it is a requirement for the cleansing mechanism.

Processing overhead of scrubbing centres is mainly due to firewalls, Intrusion Detection Systems (IDSs), in-line filtering, and complex multi-level IP traceback methods [Savage et al., 2001]. Similarly, these are proprietary defence mechanisms provided by ISPs and cloud providers. This makes them highly expensive. One such example is a report from the General Services Administration (GSA) Schedule which states that a DoS defence appliance that can defend at most 10 Gbps attack has a price tag of ~\$128,000 [GSA Store,].

Some of the problems associated with this approach are:

- If a victim is attacked often, then cloud-based solutions become financially expensive.
- An Enterprise has to trust an external entity with its data in transit. It also has to spend extra resources on security of data in motion apart from costs incurred through a cloud provider. We present ILNP-based solutions where an enterprise does not need to trust external entities.
- It requires traffic redirection, which introduces further latency.

A cloud-based security provider can also distribute incoming traffic across multiple regions where each region contains replication of the same protected service. Cloud providers make use of anycast IP addresses to achieve this scenario where an attacker using the destination anycast IP of the victim does not know which paths the attack traffic will take.

The cloud based security against DDoS attack is termed as DDoS protection as a service. The enterprise clients do not concern themselves about how they are being protected as long as the Service Level Objectives (SLOs) are being met by the DDoS security provider.

Using our defences, an enterprise has complete control over its own traffic, and they do not require an enterprise to trust an external security provider. We made use of DNS instead of BGP, where DNS is also part of the enterprise network.

2.3.7.3 In-line Filtering

This defence can use an appliance that is deployed between a server and the Internet. This appliance can inspect the packets for correct fragmentation flags, TCP/IP headers, etc. It drops the packets which violate certain security restrictions and it can also make decisions based on machine learning enabled bloom filters [Tseung et al., 2017], packet marking assisted filters [Zhang et al., 2009], Artificial Neural Network (ANN) assisted filters [Peraković et al., 2016] etc.

Some of the problems associated with this approach are:

- If DoS traffic exceeds the capacity of the link connected to an in-line filtering appliance, then this technique becomes ineffective.
- An enterprise might need to deploy multiple expensive appliances at its edge network that might require subscriptions, licences, and software updates/patches.

In this research, we evaluated defences which require a backend processor. This backend processor is an off-the-shelf Linux machine with individual components made from opensource software, and it is deployed between the DNS server and the victim hosts or the victim access routers.

Our backend processor is not a filtration appliance but it does require a packet filter at the victim host or a secure access to enterprise edge routers and its DNS. Our two (see chapters 3 and 5) out of three defences require a host-based firewall with stateless packet inspection, i.e., it does not track client session states. This host-based packet filter compares client naming information with allocated DNS capability and drops or accepts traffic accordingly.

It should be noted that this research does not provide financial cost analysis of our defences against market leading appliances.

2.3.7.4 Reputation Based Approaches

Reputation-based approaches apply a reputation metric to the network traffic and make decisions on acceptance or rejection of traffic flows based on good or bad reputation of these flows. TrustGuard [Liu et al., 2011] is an example of such a system in which a credit metric is accumulated over time for traffic that shows low packet-size distribution diversity [Du & Abe, 2008]. Traffic with low diversity, e.g., DoS, gets dropped due to a low credit score.

Reputation-based approaches might also be used in the context of an overlay network where they allow the detection of DoS attacks in a decentralized and distributed manner. One such approach is EigenTrust [Rao et al., 2010]. Typical issues in these systems are data pollution, client/server privacy, and trust [Al-Qudah et al., 2016]. Approaches similar to EigenTrust are not efficient in some decentralized networks as well [Lua et al., 2014].

In our DNS capabilities based system, a victim host gives a capability to legitimate clients through our backend for DNS, but our defence does not decide if a client, which has requested a capability, is either legitimate or malicious. It should be emphasised that our defence works in the context of spoofing-based DoS attacks.

We recommend the use of approaches such as, TrustGuard as a source of client information, in order to provide or deny DNS capability through our backend processor. It is a future work in which our backend processor can make use of an ACL that can be generated by a reputation-based system. In return, our backend processor can also provide feedback on well-behaving connections.

2.3.8 Lack Of Flexible And Elastic DoS Defences

It has been observed that appliances that are used for defence against DoS are mostly monolithic. These devices provide security for the worst case, i.e., if they are designed to handle the DoS attack of 10Gbps then it will cost the same no matter if one gets a total of 0.5Gbps throughout the attack in the lifetime of the service [Fayaz et al., 2015].

In our research, a flexible approach is taken by introducing three (see chapters 3 to 5) different mechanisms (multi-layered security) to provide security against DoS attacks. Two of these defences defend, individually, against TCP SYN flooding and UDP flooding-based attacks. The third defence simultaneously defends the victim against both TCP SYN flooding and UDP flooding-based attacks. If any one or two of the defences fail, then victim nodes have still the option of using a third one.

2.4 Security Paradigms

This section discusses two paradigms related to our research. These are Moving Target Defence (MTD) and DNS *capabilities*.

2.4.1 Moving Target Defence (MTD)

Communication systems with fixed IP addresses or system configurations with no access authorization are accessible by attackers. MTD allows a system to change such configurations as a function of time to make it difficult for an attacker to plan and launch an attack. The goal of the MTD paradigm is to increase the complexity and uncertainty for attackers while maintaining legitimate client access and end-to-end connectivity.

Chapter 1 (see §1.4) defines MTD as an adaptive security mechanism for maintaining operational integrity and the resilience of systems by randomizing information available in host/network configurations. This information can be IP addresses, TCP/IP port numbers, network topologies, ILNP

namespaces, Autonomous System (AS) numbers, network identifiers, computer instruction sets etc. Our defences make use of ILNP namespaces as a randomized information resource.

Through our approach to security, we introduce an unpredictable, and short-lived *attack surface* to the attacker with the help of MTD. An attack surface is one or more accessible vulnerabilities in a communication system. A DoS attacker can utilize an attack surface to successfully launch an attack. A static IP address, or a namespace, presents an attack surface for an attacker. Our research introduces complexity and uncertainty of host/network access for an attacker by dynamically changing this attack surface.

The MTD paradigm introduces system property metrics which every MTD compliant defence has to employ. In the next sub-sections, we discuss these metrics and their applicability to our defences.

2.4.1.1 Moving Property

Any property of a communications system which can change over time is a *moving property*. For example, if the address or namespace of a node is changed frequently then it would seem as if the node is either mobile or has been changed into a different node, even though its physical location or its application services might be fixed.

A moving property metric alone does not provide any benefit unless it is coupled with some other properties that will be discussed in the next two sub-sections. This property has the tendency to reduce vulnerability testing and random probing by the attacker because the attacker might not know the active duration of some attack surface. Vulnerability testing and random probing are two methods to profile a communications system for understanding its behaviour, and attack surface.

We will use the term *transient* or *ephemeral* for an unpredictable and frequently-changed configuration metric.

The following are the requirements for a moving property metric:

Unpredictable Future Value Of The Transient Metric:

The attacker should not be able to calculate or find the next value of the system configuration being randomized. In our defences, we chose one or two 64-bit ILNP namespace values using random seed values controlled by victim host/site. Our defences-generated random namespace values are communicated through the updates in the DNS data (for new clients) or through ILNP-specific control messages (cf. §2.1.1.2) during the communication session.

Sufficient Set-Size Of Transient Metric Values:

The value-space of this metric should be large enough so that any attacker would have to exhaust their resources to successfully predict and use a single transient value. A victim node can also choose a random value of the *time-interval* after which a new transient value can be used. In this specific case, a victim node will use two system configuration metrics (see §2.4.1.1). In our defences, we use two 64-bit ILNPv6 namespaces which provide a sufficiently large transient value-space (a pool of $2^{64} = 18,446,744,073,709,551,616$ for at least one namespace).

Periodicity Of Active Transient Metric:

The victim should effectively choose the periodicity of the transient configuration metric. The time between two distinct values of the transient configuration metric should be small enough so that the previous value should not be used effectively by an attacker.

In the evaluation of our defences, we used values equal to or less than 20 seconds. We chose an arbitrary small value because we tested our solutions on 25 client sessions, each one of at least 160 seconds under six different network conditions. This gave us enough periodicity to evaluate the feasibility of our defences. It should be noted that it is a management decision to choose the periodicity of this metric based on management requirements without affecting the MTD requirements. If an attacker can launch an attack during the 20 seconds duration, then we recommended reducing this further. Our defences make use of mechanism in which it allows very little duration (the overhead has been evaluated in earlier researches, i.e., [Phoomikiattisak, 2016], and §§3.4 and 4.2.4).

As mentioned previously, our solutions employ the NID64 and 64-bit Locator (L64) namespaces as transient configuration metrics, either individually or combined. In our research, when two metrics are combined, we get a new transient metric which promises strong node security and the possibility of enhanced client privacy as a positive side effect.

2.4.1.2 Authorization Property

The authorization property specifies that only an authorized client should be allowed to access the victim node, while the victim host or any network that is aware of this authorization can drop unauthorized packets. A mapping system can be placed in the network to map a specific moving property with a specific authorized client. If an unauthorized client accesses the victim node then a decision to accept or reject the client connection request can be made based on the available mapping. In our research, our backend processor creates a mapping and installs it either in the victim network or the victim host.

Sharing Transient Values With The Attacker:

An authorized client should not share the transient value of the MTD configuration metric (an IP address, a NID64 value, or an L64 value) with an attacker, otherwise an attacker can use spoofed sources to gain access to victim node.

In our defence for host security (see chapters 3 and 5), we used a mapping system which maps a specific client with a specific transient value of NID64, or NID64-L64 combination, to reduce the effects of information sharing. Any client which is not in the map will be disallowed by the firewall placed in the victim host. But, if an authorized client shares our granted transient value, then our defence cannot defend the victim spoofing based attack. But, we recommend that once it is detected that a given transient value was shared, then the backend processor should make the mapping void. It is also recommended to log this as a security breach incident.

In our defence for network security (see chapters 4 and 5), we change the transient value *within* the client session. It should be noted that our research mandates the use of strong authentication for authorization, but does not mandate the use of a particular authentication mechanism over another. Our research also does not provide any authentication mechanism, but it grants a transient value to an already authenticated client.

2.4.2 Domain Name System (DNS) Capabilities

A *DNS capability* is a set of bits, distributed through DNS, that allows its holder to access another domain owner for a limited time. A domain *owner name* in DNS zone files is specified as a Fully Qualified Domain Name (FQDN) where the domain owner can be a host or a network. In DNS capabilities, a client can only get a capability through DNS. A DNS server can decide whether to give a specific capability to a client or not based on multiple factors which might be the presence/absence of the client in an Access Control List (ACL) or other authentication processes. We emphasise that the authentication process is a precursor to an allotment of a capability.

All of our three (D)DoS defences use DNS capabilities. The decision to allocate a specific capability is not part of our research, so we do not provide or use any authentication process. We recommend use of strong authentication processes before a DNS capability allocation, as mentioned earlier.

Our backend processor is essentially a backend for DNS that calculates DNS capability values, maps them to firewalls, controls access router interfaces based on them, and enforces them *end-to-end*. We use the term *end-to-end* with caution and in a specific context because clients are not aware of our backend processor, they cannot get DNS capabilities unless the backend processor allocates them. Once the backend processor has allocated

a capability, it forwards it to the DNS server which carries them to the client through DNS response messages/packets. We use DNS as a distributor for capabilities because of its wide use in the current Internet. Since we use enterprise-controlled DNS, our backend processor and the enterprise hosts are not exposed to external entities.

2.4.2.1 Examples Of DNS Capabilities-Based Defences

This section provides examples of DNS capabilities-based systems along with their comparison to our approach to security.

Inexpensive Network Capabilities By Shue et al. [Shue et al., 2012]

Shue et al. proposed a method which makes use of DNS to distribute a set of temporary time-dependent IP addresses. At the victim node, there is a Network Address Translation (NAT) installation which translates the temporary IP addresses to the host-specific ones. Any client that has a temporary IP address distributed through the DNS can connect to the target through the NAT device. The term *inexpensive* only entails that it uses the DNS for the distribution of the capabilities.

In our approach, we use this mechanism with temporary ILNP namespaces rather than temporary IP addresses. We do not use NAT since there is no requirement for doing so in ILNP and the functionality is available as a dynamic namespace binding feature. We believe that the term *inexpensive* can also be applied to our defences, as they not only use DNS (as done by Shue et al.) but also eliminate the need for NAT. It should be noted that neither Shue et al., nor we, have done financial cost analysis of our defences.

The following are some of the issues associated with the method proposed by Shue et al.:

- It uses two tables in NAT. One table maps temporary IP addresses to an internal IP address. And the second table tracks established connections. The coordination between NAT tables breaks connections in some routers such as, Linksys WRT54Gv2. In our locator-based defence (see chapter 4), we do not use a firewall. In two of our defences, we use a host-based firewall with only one map to tally (see chapters 3 and 5).
- As some web browsers do not honour DNS TTL values, they might use expired temporary IP addresses. Our approaches which use ILNP's NID namespace do suffer from the same issue. But our defence mechanisms that use L64 namespaces do not suffer from this issue.
- It is primarily used and tested with IPv4. Its working with IPv6 is proposed. It does not protect the network unless there is an IP

extension mechanism (e.g., Network Mobility (NEMO) Extensions for Mobile IPv4, IPv6 Mobility, etc). Our approaches which use an ILNP's L64 namespace can protect the victim network as well as it natively support host/network mobility.

Shue et al.'s mechanism takes 13 milliseconds to update an entry in the DNS server and 4 milliseconds to add a NAT rule. There is no publicly available implementation of the Shue et al. mechanism, so we were unable to perform any kind of empirical evaluation that compares ILNP capabilities. We should emphasise that there was no need for us to compare ILNP based capabilities against IPv6 based mechanism from Shue et al. because our objective was to compare our approach to SYN Cookies. We recommend a future evaluation where ILNP based mechanism should be compared with a wide variety of IPv6 based capabilities.

Evasive Internet Protocol (EIP) [Al-Qudah et al., 2016]

In this approach, DNS is used to distribute capabilities. It emphasises that the DNS infrastructure should use EIP as well. It also mandates that only the root DNS servers may not be using the Evasive Internet Protocol (EIP) because DNS root servers are already highly distributed, replicated, and accessed through anycast addresses. Our defences do not require DNS servers to be updated but they are used as forwarders.

EIP requires support from the The Internet Corporation for Assigned Names and Numbers (ICANN) to bootstrap certificates attesting the ownership of IP addresses. Similarly, it requires support from DNS root servers to offer EIP protection to secondary-level domain nameservers. It uses the Resource Public Key Infrastructure Framework (RPKI) to secure capabilities distribution. Its DNS capabilities are based on the transient IP addresses, capability validity constraints, and capability issuance timestamps. The validity constraints make use of two aspects, the capability lifetime, and the allowed number of bytes that can be sent within this lifetime. Any packet that does not conform to these validity constraints gets dropped by the EIP-supported devices which can be on-path routers, or the destination network or a host.

2.4.2.2 Non-DNS Based Capabilities

Secure Architecture for the Networked Enterprise (SANE) [Casado et al., 2006]

Secure Architecture for the Networked Enterprise (SANE) uses network security policies to generate switch-level source routes to act as capabilities. These source routes are encrypted at each switch level using keys for each switch. This information is carried in the SANE header that resides in the

Ethernet frame and thus works at the link layer. These security policies are topology-independent.

SANE uses a protection layer which is used for the communication among end hosts and the protected hosts. This protection layer is similar to a mapping layer. The main component of the protection layer is the Domain Controller (DC). The DC authenticates the users as well as the hosts. It also advertises the services that are offered by the protected network. It also decides who can connect to these services. It uses symmetric keys to distinguish users and hosts. The information about addressing and reachability is contained in the DC itself using a Network Service Directory. This eliminates the use of DNS for service discovery.

NetFence [Liu et al., 2010]

NetFence uses a secure congestion policing feedback in the network routers, which stamp the packets for feedback mechanism, to control the congestion channels. They police the links so that these links can be monitored and acted upon in the face of an attack. Routers update the feedback in the packet headers. The end systems use the aforementioned feedback information in the packet headers as a form of capability token to control traffic. The same system is used for rate-limiting at the router level as well.

The following are two important issues with this defence:

- Malicious on-path routers can drop the feedback packets and can damage the end-to-end token states used by the victims.
- It requires the routers at the congested links and the access routers to be modified. Our approach does not require any modification in the external networks, but it does require victim access routers to be controlled by our backend processor.

Traffic Validation Architecture (TVA) [Yang et al., 2005] [Yang et al., 2008]

The Traffic Validation Architecture (TVA) uses an approach in which the packets contain capability information given to them by the victim. On-path routers are modified so that they can inspect capabilities, and if the capability does not match the path it is traversing, then they are dropped at the access level.

In this approach, each packet carries this information. The capabilities are granted through piggybacking the TCP SYN/ACK packets with capabilities. Each edge router stamps the capability request packets with a 16-bit unique tag. This tag is only used to identify upstream providers. A path identifier is built using these tags by appending them together at every upstream level. The routers maintain queues with path identifiers and compare them to each packet that contains a path identifier.

TVA is vulnerable to attack by an attacker who uses fake path identifiers and tries to congest the routers' queues.

2.4.3 Denial Of Capabilities Attacks

Denial of Capabilities attack a target's capability request channel [Argyraiki & Cheriton, 2005a]. Capability request packets do not carry capabilities so they are allowed to reach the capability distribution node. This form of attack can cripple any capabilities-based defence.

Our research does not provide solutions for Denial of Capabilities attacks, but since our backend processor is within the protected enterprise network and its location is a secret, there is less possibility, unless an attack is launched from within the enterprise, of such an attack on the backend processor. The enterprise DNS is still vulnerable to DoS attacks, so indirectly our capability request channel is compromised in such attacks. We recommend that an enterprise uses a redundant DNS infrastructure with multiple points of entry to secure the capability request channel.

2.4.3.1 Defence Against Denial Of Capabilities Attacks

Some of the proposals to deal with this problem are given below:

Distributed Cryptographic Client Puzzles:

In this solution, the client who is requesting the capability is asked to perform a computationally-intensive cryptographic puzzle. The calculation is derived in such a manner that the effort required by the victim to check the results is far less than the effort required by the client to generate a valid result. Portcullis [Parno et al., 2007] uses client puzzles.

Client puzzles are used to differentiate among good and bad hosts. Current DoS attack tools do not employ any technique to inform a victim about its usage in malicious activities. Each rejected client is asked to solve a higher difficulty level puzzle. The capability channel is unreachable to those clients which fail to perform the puzzle.

Rate Limiting The Capability Channel

In this approach, there are multiple ways to reach the capability distributor. If there is a rate limiting on each entry point to the distributor then, if there is an attack on one link, all other clients on the other links will get the capability, and those accessing this congested channel will shift to other low congestion channels. To do a successful attack, the attacker then has to attack all the links that lead to the distributor.

Using The DNS Infrastructure:

EIP (see §2.4.2.1) uses the DNS infrastructure to help protect the capability distribution channel. The DNS is also used for capability distribution and it also uses the EIP protocol. The scope is now narrowed to the root DNS servers. Since the root DNS infrastructure is highly distributed, the attacker has to expend a lot of effort to dismantle the capability distribution service.

2.5 Naming Based Protocols And DoS Defence

Naming based protocols semantically separate a node's identity and its location. This is not the case with IPv6 where the node identity and its location are tied to the same 128-bit IP address. Some of the naming-based solutions are ILNP, Evolution [Khare et al., 2010], Renumbering [Carpenter et al., 2010], Locator/ID Separation Protocol (LISP) [Farinacci et al., 2013], Host Identity Protocol (HIP) [Moskowitz & Nikander, 2006], Routing Architecture for the Next Generation Internet (RANGI) [Xiaohu Xu, 2010], Name Overlay (NOL) [Wang et al., 2010], etc. We will only cover LISP, and HIP as we have found DoS defences using them in the literature.

2.5.1 LISP And DDoS Defence

[Luo et al., 2013] argues that LISP can be used as a DDoS defence mechanism. An identifier-to-locator mapping approach is proposed. This approach makes it difficult for the attackers to control the botnets, hence controlling the DDoS attack itself.

It argues that the same approach can be used to detect a DDoS attacks. The main idea for DDoS detection is that, since each victim server's locator is queried from the DNS through the victim server's identifier, if we count the frequency of these requests then we can make a metric which shows that a DDoS attack is in progress.

The trace-back also becomes easy since the Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) know each other when a packet arrives at their location.

The DDoS attack can be prevented through knowing that if the attacker wants to send the command and control messages to the zombie, then it has to know the locator of that zombie. If the zombie is a non-server, then ITR will not reply with a locator value. This way the zombie will be unreachable. If the zombie is a server, then the attacker can get the locator value and pass on the commands. Since the number of servers in the Internet is a fraction of the number of non-server systems, the LISP-based mechanism is able to control the flow of commands from the attacker.

2.5.2 HIP And DDoS Defence

[[Liyana et al., 2015](#)] shows how HIP can be used to create a secure environment in software-defined mobile networks. It utilizes both HIP and IPSec. The basis for such approach is the HIP RFC 7401 [[Moskowitz et al., 2015](#)] which specifically designs the protocol with DDoS in consideration. The authors of [[Liyana et al., 2015](#)] use HIP to form secure tunnels between the Data Plane switches and the controller.

2.5.3 ILNP And DoS Defence

ILNP provides identifier and locator namespaces for the node identification and its location. A single host can have multiple locators with the same identifier, meaning that a transport layer session can use multiple locators without disrupting communications.

The *capability*-based solutions demand that there should be one or more network properties/configurations/components of the system which should continuously change. Since ILNP has two crisply separated namespaces that can act as transient properties/configurations/components which identify a node on a specific location in a network, we can employ these namespaces to further study their effectiveness in a DoS defence solution. This work is the first of its kind that investigates the feasibility of ILNP namespaces to be used in a DoS defence.

2.6 Security Challenges And Solutions Matrix

Security from malicious network-based on-path or off-path attacks is a requirement in modern times, because we rely on the Internet for every aspect of our lives from acquiring knowledge and doing business, to making financial transactions, etc. It also entails that those businesses which provide these services to Internet users should have online services running and available at the time when they are required to run and facilitate. For these reasons, DoS attacks are quite common and new network functionality should also provide solutions to this dire problem.

2.6.1 Network Reconnaissance Attacks

These are monitoring-based attacks which gather information about a victim. This monitoring leads to planning and execution of certain attacks on node/site availability. Network reconnaissance is one such aspect of pervasive monitoring, whereas pervasive monitoring itself is perceived as an attack [[Farrell & Tschofenig, 2014](#)].

The Internet Engineering Task Force (IETF) recommends that the design of new protocols should cater for solutions to pervasive monitoring.

Pervasive monitoring often yield private information and can also lead to breach of security. The challenge is that the design of the solution should ensure that the attacker should have a hard time to mount a successful reconnaissance attack. It should be expensive and infeasible for the attacker.

DDoS have become quite common, and easy to mount. There are many solutions to this but most of them are complex in nature and require considerable effort. In this research, we make an effort to mitigate such attacks.

2.6.2 ILNP-based Security Solutions

The solutions provided in literature lack the use of naming, and in-fact there is no single evaluated DoS defence solution that is based on ILNP. Since ILNP uses two different namespaces for addressing and most of the IPv6-based solutions are based on the addressing, it is feasible to test the ILNP-based naming to come up with better solutions than are provided through IPv6.

2.6.3 The Role Of DoS Fail-over

The DoS fail-over is a process of applying a mechanism which effectively takes a victim out of the disruption caused by DoS attack. Ideally the sessions should be maintained after the fail-over and the data-rate should be improved as well. The sessions can be maintained in ILNP (but not possible in IPv6) case due to its dependence on NID namespace which is only used at the transport layer. In case of a reduced data-rate due to disruption, the fail-over should improve the data-rate that should satisfy the Service Level Objectives (SLOs) of the enterprise. This fail-over has to be quick, and is one of the most important parts of certain attack mitigations.

We are going to use this fail-over to create a mechanism which hides the identity and/or location of a victim. Since the identity and/or location of the victim will be unknown to the attacker, it would be hard for the attacker to mount a successful attack.

ILNP provides a mechanism which can be used as a network/node fail-over in the context of a DoS attack. ILNP, natively, does not prevent the attack, but it does provide semantically-unique naming which makes a fail-over possible with minimal session disruption. Our mechanism will not affect the native services provided by ILNP.

2.7 Summary

This chapter presented evidence of the problem domain while also covering the state of the art of DoS attacks. DoS attacks are classified in terms of the attack vectors which can be used to launch an attack. We mainly covered low-rate TCP SYN flooding and traffic-based volumetric attack types.

The architecture of ILNP and its required functionality that can be used to mitigate DoS attacks was also covered. An ILNP's mobility mechanism, which has been extensively evaluated in early researches, is used to mitigate volumetric traffic-based attacks. ILNP provides a crisp (loose-coupling) separation of identity and location of a node. This feature can be used to form defences which are not possible in the native functionality of IPv6. Our main focus is to mitigate DoS attacks and not to empirically compare ILNP features with those of IPv6.

The DNS infrastructure element which we used in all of our defences has also been described along with operational details, support for ILNP namespaces as resource records, and its relevance to support enterprise security.

This chapter also covered the state-of-the-art security paradigms, i.e., DNS capabilities, Moving Target Defence (MTD), and DNS fast flux. All of our defences make use of these paradigms, DNS, and ILNP to support enterprise security both at a host-level (§1.3.1) and a network-level (§1.3.2).

Chapter 3

Protecting Enterprise Hosts From Transport Layer DoS Attacks

An enterprise runs diverse mission critical applications whose availability at a required times is of importance to its business goals. Enterprises have to ensure that their systems are available when external clients need them. SYN floods attack the availability of such systems §2.3.2.1.

Our goal, in this chapter, is to propose and evaluate a defence against transport layer SYN flood attacks. SYN floods affect enterprises of all levels by exploiting TCP connection establishment.

We present an evaluation of how ILNP and DNS can be used to mitigate SYN flood attacks. We created a solution which is expected to mitigate such attacks by comparing it to SYN Cookies §2.3.6.1.

This chapter covers the following three parts:

- Establishing a rationale for investigating an ILNP-based solution against TCP SYN flood attacks. §2.3.2.1 gave a thorough description of what TCP SYN floods are, and §2.3.6 provided current mitigations. In this chapter, we will also present shortcomings of current mitigations. Definitions of new concepts, and relevant conditions/limitations in the applicability of these concepts will also be presented.
- An empirical comparison of ILNP-based defence and SYN Cookies using a set of experiments in varying network conditions. Each conditional evaluation will be referenced against a baseline which consists of client communication during an attack but without any solution.
- Empirically evaluating the performance of end-to-end DNS request/response signalling in the presence and absence of ILNP-based defence.

3.1 Rationale

RFC-4987 (TCP SYN flooding attacks and common mitigations) [Eddy, 2007] documents some problems with TCP SYN Cookies which are widely deployed in current Internet. In the following, we document the shortcomings and disadvantages of SYN Cookies:

- SYN Cookies introduce disruptions in unidirectional data flow if Selective ACK (SACK) blocks are used. So the SACK block usage is discouraged with SYN Cookies, whereas SACKs enhance TCP performance.
- Generating SYN Cookies is problematic. Solutions for such problems are only implemented in FreeBSD.
- SYN Cookies are incompatible with application data piggybacking over TCP SYN packets.
- SYN Cookies introduce packet loss in application data when the data originates from a passive host, e.g., from Simple Mail Transfer Protocol (SMTP) application servers.
- SYN Cookies are not recommended to be used in current and future TCP extensions.
- SYN Cookies are incompatible with TCP options.
- SYN Cookies hamper correct error reporting in Linux kernels as documented in Linux Kernel Driver DataBase (LKDDDB) [LKDDDB, 2018].
- SYN Cookies introduce inconsistency problems in IPv6 Flow Label across a single IPv6 session [McGann & Malone, 2006].
- SYN Cookies introduce Application layer packet loss which is empirically evaluated in [Cole, 2018].

We introduce a new network-based solution to SYN flood DoS attacks using ILNP's NID64 namespace and DNS. Our defence does not suffer from any of the problems mentioned above, because a SYN flood cannot reach the victim host network stack while our defence is running, thereby eliminating the need for SYN Cookies.

3.2 NID64 Capabilities (NC64)

We introduce the concept of NC64 defence. The new components in enterprise hosts and network that are required to enable NC64 are shown in

Figure 3.1. NC64 defence only runs within an enterprise network. A NC64 defence is enabled using NC64 capabilities. A NC64 capability is a specific form of a 64-bit identity of a publicly accessible victim host which is given to an authorized external client of an enterprise for a short duration. Any unauthorized external client can also reach the victim network, but it cannot reach services running on the victim host.

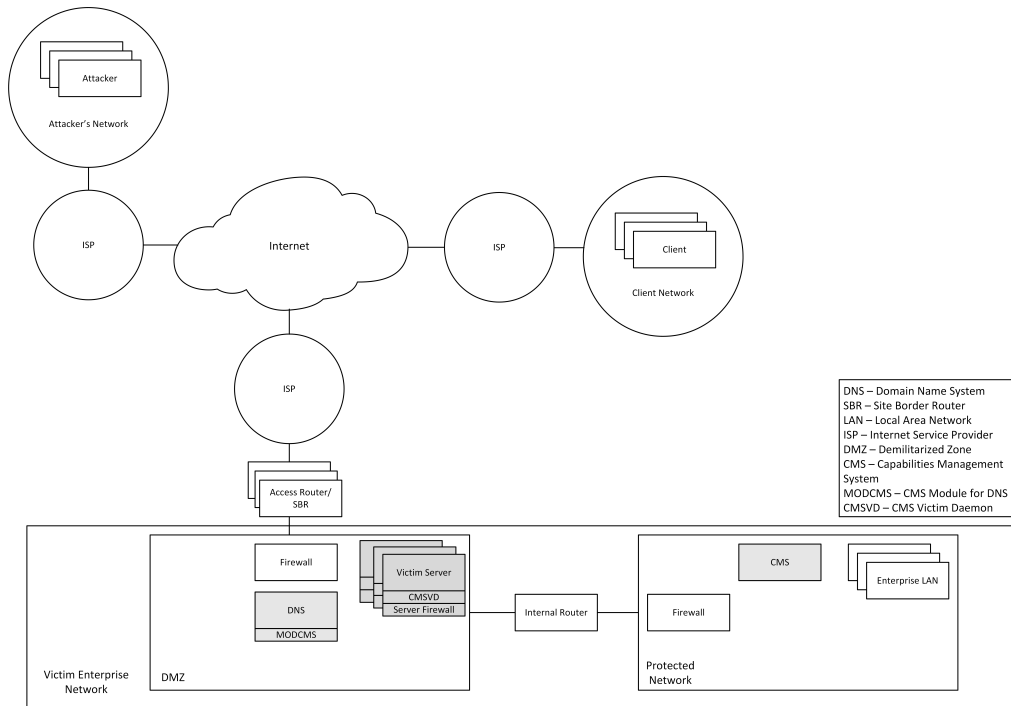


Figure 3.1: Required modifications in enterprise network for NC64 defence

3.2.1 New Network Components In NC64 Defence

There are three new components in Figure 3.1. Definitions, responsibilities, and implementation details of these components are given below:

3.2.1.1 Capabilities Management Server (CMS)

CMS is our custom C implementation of the backend processor for DNS. It gets capability requests from MODCMS which is a DNS module (see §3.2.1.2), calculates capabilities, installs them on the publicly accessible victim host, and then returns capability responses back to the DNS module.

CMS is protected by the Demilitarized Zone (DMZ) of the enterprise network, i.e., it runs in the protected network. Identity and location of the CMS is a secret (it cannot be found by scanning from the external network of the enterprise) shared only with the enterprise-managed DNS module and

publicly accessible victim hosts. CMS can run on IPv6 or ILNP kernel with off-the-shelf server hardware. We run the CMS on Ubuntu Server 16.04.

3.2.1.2 CMS Module For DNS (MODCMS)

MODCMS is a DNS server software module which can be in a separate machine than the machine running the DNS software, but we have collocated it with DNS software. MODCMS is implemented in C by us, and we run it on Ubuntu Server 16.04 with off-the-shelf server hardware. It gets DNS requests from DNS software, extracts client addresses, creates a TLS encrypted capability request message, and sends it to CMS over the secure channel. Upon reception of capability response from the CMS, MODCMS creates a DNS response message and forwards it to the DNS software which later can forward the DNS response to the client.

MODCMS does not communicate with any other entity but the CMS and DNS. Its identity and location are secret shared between DNS software and CMS software.

3.2.1.3 CMS Victim Daemon (CMSVD)

CMSVD is our custom C implementation which only communicates with CMS on a TLS encrypted channel. It receives instructions from the CMS to create firewall rules for each NC64 capability, and returns acknowledgements to CMS. It can run on off-the-shelf server hardware. The CMSVD daemon is independent from other software applications running on the victim host. We are running it on an Ubuntu Server 16.04 with off-the-shelf server hardware.

It has a direct link with the CMS, and it does not interfere with applications running on the victim.

3.2.2 Properties Of A NC64 Capability

The binary value of a capability and its lifetime is a management decision. In our proposed design, CMS acts as an enforcer of such management decisions.

NC64 is returned to the client by CMS, through DNS, on behalf of the victim server. It authorizes the client to initiate a connection with the victim for a short duration owing to its ephemeral nature.

In ILNP, each NC64 capability has a fixed L64 value but the NID64 might change for each DNS request. It is a management decision to either make use of a different NC64 capability per DNS request per client or only use a single NC64 capability for a particular client who is making multiple DNS requests.

It is recommended to use a unique NC64 capability for well-behaved clients for an extended duration. A well-behaved client is a client who is not a security or privacy threat to a victim. An enterprise can make a decision

on who is a well-behaved client based on past traffic patterns of the client, or lack of authentication.

Clients that do not own a NC64 capability are not allowed to start an ILNP session with a victim host at the physical layer, though they can reach a victim network since the value of the L64 namespace is fixed. We will show in chapter 4 that even the network path is restricted to unauthorized clients by using L64-based capabilities.

A NC64 capability will not change for the duration of a TCP Connection. The client is allowed to initiate multiple TCP connections using the same NC64 capability. Although our solution is flexible to have mappings (see §3.2.3) with port numbers as another decision point for a different NC64 allocation, meaning each service can have its own NC64 capability or a set of NC64 capabilities per service.

A NC64 capability cannot be shared among other clients, which means that any DoS attack will become ineffective even if the attacker shares unauthorized identities among diverse entities. This case is only valid if the MAP contains information about the originating network of the client. If an attacker is in non-originating networks, it will not be able to affect the victim's services since the spoofed address is tied to a different network within the MAP.

3.2.3 NC64 Defence Capability Mappings

We introduce the concept of an NC64 mapping, hereby termed as a MAP. A MAP is a pairing of an ephemeral NC64 value and naming information of the client. For example, a MAP can be formed using a client's identifier/locator namespace and the victim's ephemeral NC64. Similarly, a service port number, a client's originating AS number, etc. can be attached to a MAP. The more client information we attach to a MAP, the more powerful a defence we can achieve. Terms like MAP, mapping, and pairing(as noun) are interchangeable.

A MAP is sent from CMS to CMSVD which implements it in the firewall of the victim host. If the firewall rule installation is successful, then a MAP acknowledgement message is returned from the CMSVD to the CMS. Otherwise, CMSVD will send an error condition to the CMS. The CMS logs this event. The client can make further DNS requests to get hold of a new capability.

3.2.4 Design For A NC64 Defence Implementation

We required a component which could calculate NC64 values and assign them to victims. For this component we designed CMS. Similarly, the requirement of having a communication between DNS and CMS, we needed to have a component to do so, as DNS software itself is not designed to com-

municate apart from the client. For this reason, we created MODCMS as a component which interfaces with the DNS and CMS. Similarly, we required a component within the victim host which should be able to understand the mappings dictated by the CMS. We termed this component as CMSVD.

For performance reasons, we collocated MODCMS with the DNS, and CMSVD with the victim host. To show that CMS can act as a middleman between DNS infrastructure and victim hosts, independent of the number of DNS servers and victim hosts, we put it as a separate machine.

The design honours the ILNP specification using mappings that ensure that a single client has a unique ephemeral destination address that remains same for the duration of the TCP session (for session continuity purposes). This state is held at the CMS and victim host (using the CMSVD and firewall).

3.2.5 Finite State Machines (FSMs) Of NC64 Defence Components

3.2.5.1 FSM for CMS

A Finite State Machine (FSM) for the CMS software daemon is shown in Figure 3.2

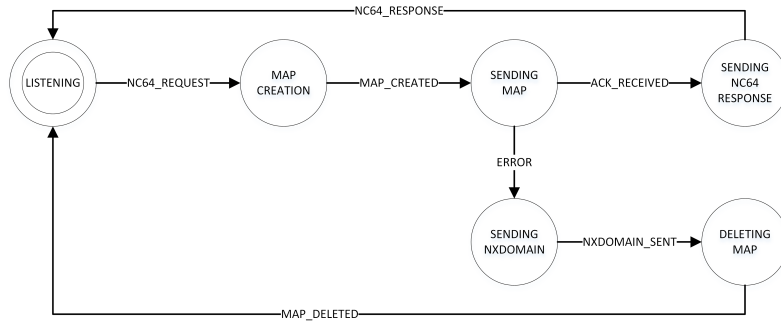


Figure 3.2: The Finite State Machine (FSM) for the CMS software

Where

- LISTENING is a state where CMS is listening for NC64 requests from MODCMS.
- MAP CREATION is a state where CMS creates a new NC64 capability.
- SENDING MAP is a state which is executed after a new NC64 capability is created. In this state, it sends the new MAP to CMSVD.

- If CMS receives an ERROR condition from CMSVD, it goes to the SENDING NXDOMAIN state where it sends the NXDOMAIN message to MODCMS which in turn delegates the NXDOMAIN response to DNS. Then DNS can send an NXDOMAIN DNS response back to the client. Once the NXDOMAIN message has been sent to the MODCMS, the CMS goes to the DELETING MAP state.
- In the DELETING MAP state, the CMS deletes its local MAP. Once it has deleted the local MAP, it then goes to the LISTEN state again.
- If CMS receives a MAP acknowledgement from CMSVD, it goes into the SENDING NC64 RESPONSE state where it sends the capability response message to MODCMS.
- Once a capability response has been sent to the MODCMS, the CMS goes to the LISTEN state again.

and,

- NC64_REQUEST is a transition when a NC64 capability request comes from MODCMS.
- MAP_CREATED is a transition when the CMS successfully creates a MAP.
- ACK_RECEIVED is a transition where the CMS receives an acknowledgement from CMSVD that a MAP has been successfully installed in a firewall of a victim host.
- ERROR is a transition where CMS receives an error message from CMSVD that there was a problem in firewall rule installation.
- NXDOMAIN_SENT is a transition where CMS successfully sends an NXDOMAIN message to MODCMS.
- NC64_RESPONSE is a transition where a NC64 capability response message has been successfully sent to MODCMS.

3.2.5.2 FSM for MODCMS

The Finite State Machine (FSM) for the MODCMS software daemon is shown in Figure 3.3

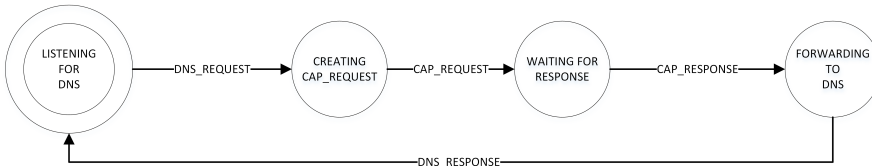


Figure 3.3: The Finite State Machine (FSM) for the MODCMS software

where,

- The LISTENING FOR DNS state listens to DNS requests.
- After receiving a DNS domain resolution request, it goes to the CREATING CAP_REQUEST state where it extracts client information, creates an NC64 capability request, and securely sends it to a CMS.
- Once a NC64 capability request has been sent to a CMS, it goes to the WAITING FOR RESPONSE state.
- After receiving the NC64 capability response from a CMS, it goes into the FORWARDING TO DNS state where it extracts the NC64 capability and the client address/identifier/locator, and forwards it to the DNS server which responds to the client with a DNS reply.

and,

- DNS_REQUEST is a transition when MODCMS receives a forwarded DNS request from DNS.
- CAP_REQUEST is a transition when a NC64 capability response has been sent to the CMS.
- CAP_RESPONSE is a transition when a MODCMS receives a valid capability response containing a valid NC64 capability and a client address/namespaces.
- DNS_RESPONSE is a transition when a MODCMS successfully forwards a DNS response message to a DNS.

3.2.5.3 FSM for CMSVD

The Finite State Machine (FSM) for the CMSVD software daemon is shown in Figure 3.4

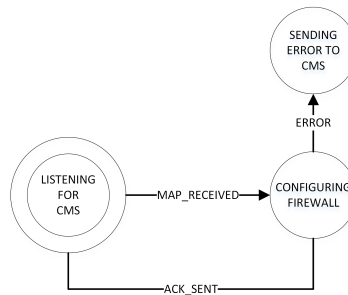


Figure 3.4: The Finite State Machine (FSM) for the CMSVD software

where,

- The LISTENING FOR CMS state listens to MAP installation requests from CMS.
- After receiving a MAP from the CMS, CMSVD goes to the CONFIGURING FIREWALL STATE where it adds the MAP to a firewall.
- If CMSVD encounters some error during the installation of the firewall rule for the MAP, then CMSVD sends an error message containing the reason. The error condition might be either that the MAP is malformed or the firewall is busy.
- After configuring the firewall, the CMSVD sends an acknowledgement message to the CMS confirming MAP installation. Once it has successfully sent an acknowledgement message, it goes back to the LISTENING FOR CMS state.

and,

- MAP_RECEIVED is a transition when the CMSVD receives a MAP installation request from CMS.
- ERROR is a transition when a NC64 MAP installation was unsuccessful.
- ACK_SENT is a transition when MODCMS successfully sends a MAP acknowledgement to CMS.

3.2.6 Defence Protocol

The following ordered steps are involved in an NC64 defence protocol:

1. A client sends a DNS name resolution request to a DNS forwarding server (DNS-F).
2. The DNS-F forwards the client request to a MODCMS.
3. The MODCMS extracts the client address/namespaces, creates a NC64 capability request, and sends it to a CMS.
4. The CMS creates a MAP and sends it to a victim server as a MAP installation request.
5. The victim server installs the MAP in its firewall, and then sends out a MAP acknowledgement message to the CMS.
6. Once mapping is acknowledged from the victim, the CMS sends the currently valid ephemeral NC64 capability as a NC64 capability response message to MODCMS.

7. MODCMS creates a DNS response message using new NC64 capability, and sends it to DNS.
8. The DNS sends a DNS response packet to the client containing an ephemeral NID64 value and a fixed L64 value.
9. The client can now initiate a data communication session with the victim using the ephemeral (or temporary) capability and the fixed L64 value.

Figure 3.5 shows a sequence diagram for the steps mentioned above.

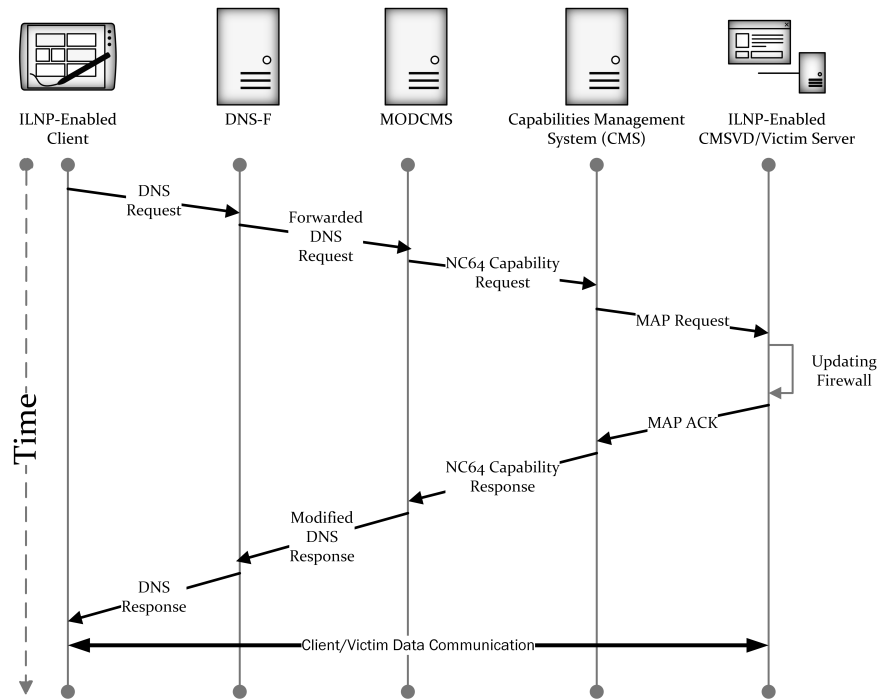


Figure 3.5: NC64 Defence Protocol Sequence Diagram

3.3 Empirical And Comparative Evaluation Of NC64 And SYN Cookies

3.3.1 Methodology And Experiment Design

There has been empirically extensive research on ILNP's mobility, and multi-homing features [Phoomikiattisak & Bhatti, 2015], [Bhatti & Atkinson, 2011], [Bhatti et al., 2016], [Phoomikiattisak, 2016], [Simpson, 2016], but there is a lack of empirical evidence that it can be used with DNS against SYN flood attacks. We designed our experiment by taking into consideration

the requirements, limitations, assumptions, tooling, testing, and statistical significance. Our null hypothesis for the NC64 defence evaluation is:

- *NC64 defence against SYN flood DoS attacks performs similar (in terms of allowing and maintaining a number of TCP connections) to SYN Cookies.*

We will test the aforementioned null hypothesis for LAN, MAN, and WAN environments, where MAN and WAN environments are emulated by introducing 25ms and 210ms end-to-end delays in the transmission path. We also chose 5% and 10% packet loss (using non-uniform normal distribution) in each direction to check how individual defences perform in the face of an extreme network congestion. Such delays and packet losses can also give us information on the behaviour of our backend.

The following are the three sub-experiments for a comprehensive evaluation (investigation of each by examining the respective client to server traffic):

1. Client to server communication during an attack with SYN Cookies and NC64 disabled
2. Client to server communication during an attack with SYN Cookies enabled
3. Client to server communication during an attack with SYN Cookies disabled but NC64 enabled

Each sub-experiment was run 25 times, with an individual runs of 5 minutes. The number of runs was chosen based on statistical power analysis which showed that at-least 21 runs would be enough to get data which can be used to extract further statistical insights. We chose 5 minutes per run because it would be enough to cater for TCP's slow start mechanisms [Sikdar et al., 2001], stable rates of attack traffic and minimization of system level affects.

Tcpdump¹ was used to capture packets and analysis was done using wireshark and R² scripts.

Figure 3.6 shows the logical diagram of the emulated enterprise network along with all the required entities for this experiment.

¹https://www.tcpdump.org/tcpdump_man.html

²<https://www.r-project.org/>

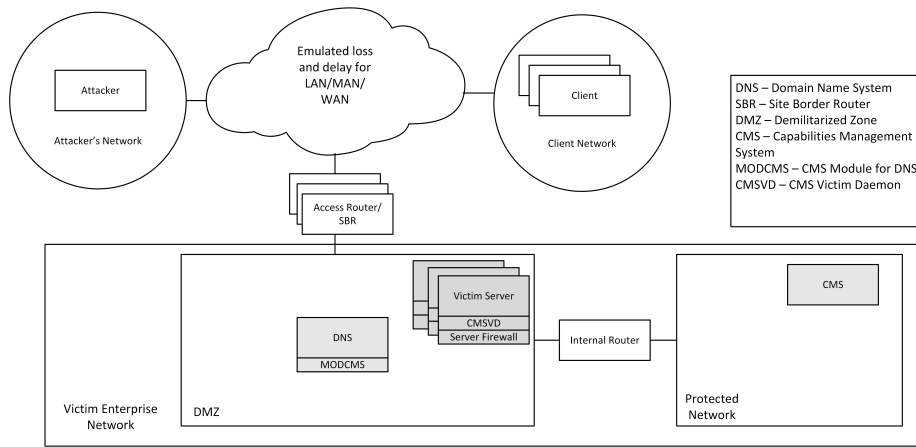


Figure 3.6: Logical network diagram for NC64 defence against SYN flood attacks. CMS identity and location is a secret to the DNS and victim servers, so it is shown as a part of the protected network. The DNS and victim server is accessible by external networks, so they are shown in the DMZ. Network emulation is done at the routers which attach the enterprise network to the client and attacker networks. There are 2,000 clients (see §3.3.1) so we show them as stacked boxes within the client network.

Baselines

SYN flood mitigation through SYN Cookies is the baseline for NC64 based defences, and defence-less client communication is the baseline for SYN Cookies. Our purpose is not to reject any alternative approach (SYN Cookies) but to establish a new approach that does not suffer from the shortcomings of SYN Cookies.

Testbed

The testbed consists of a DNS forwarding server, a victim machine running an ILNP kernel, a CMS server, an attacker machine, and 2,000 virtual clients (hosted by a single dedicated client machine running an ILNP kernel).

The setup consists of a control virtual Local Area Network (vLAN) and a data vLAN. The control vLAN is used to administer the execution and collection of results. The data vLAN is used for running the actual experiment. Figure 3.7 shows our testbed.

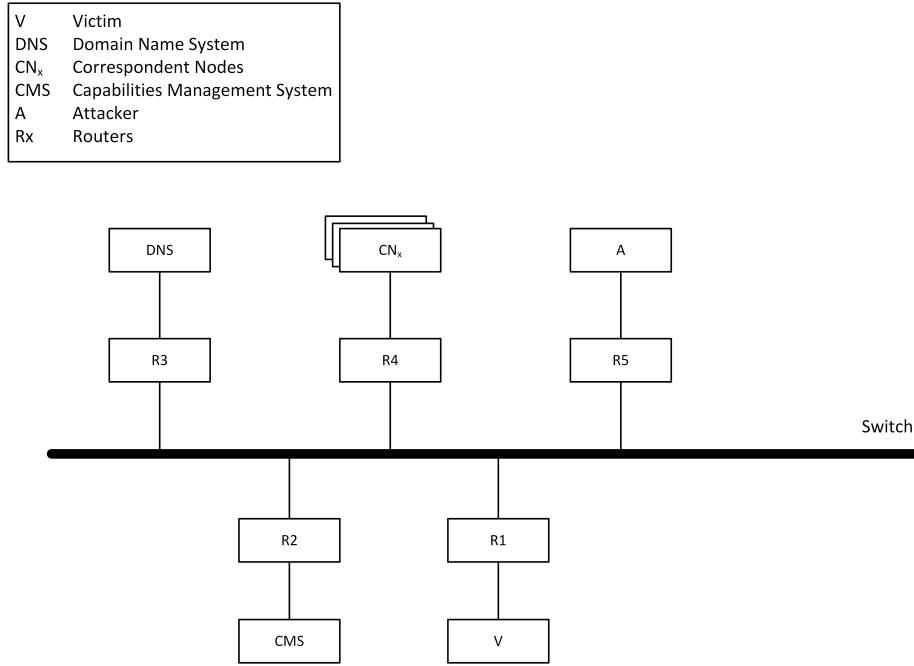


Figure 3.7: A testbed for NC64 based defence against SYN flood attacks. Each entity that is attached to a router is part of a separate network. Each router is an unmodified Linux box with packet forwarding enabled. Each router can add packet delay to emulate MAN and WAN environments based on the needs of an individual sub-experiment. This topology is chosen so as to model a real world enterprise network. Each router is creating a separate network (external or internal). R4 and R5 act as edge routers.

The server machines and the routers, in the testbed, run Ubuntu Server 16.04.

Each router is directly connected to an Extreme[®] Switch x45a-48t with an isolated port, using 1 Gbps full-duplex Ethernet links. Each machine, apart from the switch, in the testbed is a Gateway[®] GR380 F1 machine with 64-bit Intel[®] Xeon[®] 8-core CPU (2.27 GHz base frequency)³.

Experiment Configurations

Knotdns, v2.4.1 is used as a DNS server software which runs the *modcms*, as a Knotdns module extension, to communicate with the *CMS*. The victim machine runs a *CMSVD* daemon which derives firewall rules in the Linux kernel using a *nftables*⁴ firewall. A *CMS* daemon runs on a separate Linux

³https://ark.intel.com/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-GHz-5_86-GTs-Intel-QPI

⁴<https://wiki.debian.org/nftables>

machine.

MAPs are communicated between the CMS and the victim server through the use of a binary serialization format-based protocol called colfer⁵. All communications are also encrypted using TLS [Dierks & Allen, 1999].

CMS, modcms, and victim daemons are programmed in C by myself, through the supervision and guidance of my supervisor.

The victim runs stable NGINX version 1.15.1⁶ HTTP server. It serves a small static page which fits in a single HTTP 200-OK packet. The client uses curl⁷ software to make HTTP requests. Multiple virtual clients are created using secondary interface addresses, c-ares⁸, and curl. These virtual clients were configured to send client traffic as fast as possible so as to congest the 1 Gbps link with client traffic. This translates to have full utilization of the server access link, i.e., the enterprise service by the virtual clients. The attacker uses thcsyn6⁹ software, which is an industry tested tool used heavily in Kali¹⁰ Linux to launch a SYN flood attack. Tcpdump¹¹ is used to collect data at the client side. Since we are concerned about HTTP 200-OK packets which show that TCP connections were made (a predictor of the victim's engagement), we can collect this information on any other system other than the victim. We chose the client, but another on-path machine can be chosen.

The victim runs an ILNP kernel version 4.9.38 for the sub-experiment that employs the NC64 defence. It runs an IPv6 kernel version 4.9.38 for the sub-experiment that employs only SYN Cookies defence. It is important to note that we are not comparing SYN Cookies performance under two different kernels but we are comparing IPv6-based SYN Cookies defence and ILNP-based non-SYN Cookies defence. It should also be noted that SYN Cookies can be enabled in either an IPv6 or an ILNP kernel without modification.

Performance Metric For Evaluation (Testing)

The number of HTTP 200-OK responses from the victim server to the client is the metric that is used to measure NC64 defence effectiveness. Here, the notion of effectiveness has two meanings. One implies that each client should get the expected response for each request that it made, and the other implies that the victim was able to work properly (serve TCP connections in the face of an attack) even when SYN Cookies were disabled.

⁵<https://github.com/pascaldekloe/colfer>

⁶<https://nginx.org>

⁷<https://curl.haxx.se>

⁸<https://c-ares.haxx.se>

⁹<https://tools.kali.org/information-gathering/thc-ipv6>

¹⁰<https://www.kali.org/>

¹¹<https://www.tcpdump.org/>

- HTTP 200-OK messages: This metric conveys information about the successful responses that are delivered at the client. As we are seeing it from a client's perspective, we have measured it at the client machine. Here, the distinction about measurement location is important. If we were to measure the RTT of individual responses then we would have measured it in the path between client and victim; otherwise it suffices to measure 200 OK messages at the client machine itself. This metric conveys complete information that the client was able to initiate, establish, and tear-down (or terminate) a TCP connection. From the victim's perspective, it means that the victim was protected from the SYN flood.

Limitations And Validations

Our designed testbed does employ real world technologies but only in the laboratory environment. While it is also important that this solution should work in a non laboratory environments, we did an emulation of such an environment. So, to satisfy such cases, we used loss and delay parameters for LAN, MAN, and WAN scenarios. The emulation is done using the *netem*¹² emulator which is well tested software to approximate production environments.

While it is important to know whether NC64 defence scales or not (as a SYN Cookies-based solution does), we did not try to scale it above 2000 virtual clients as we have a limited number of machines in the laboratory. We chose 2,000 clients as it was the maximum that can be achieved on the machine that was available in the testbed, without affecting victim functions. For reasons of our performance metrics as mentioned earlier, it was reasonable to use virtual clients rather than physical clients because we are measuring HTTP 200-OK responses and not the RTT characteristics of such responses.

We also validated our research using two paths within the testbed. One path runs control traffic and the other path runs data traffic as mentioned in all of our experiments. This makes sure that we achieve separation of concerns and do not introduce bias in results. Given these controlled environments we were able to reproduce experimental results, 25 times.

3.3.2 Results

3.3.2.1 LAN Environment

Figure 3.8 shows the results for SYN Cookies showing an average of $\sim 24,000$ HTTP 200-OK responses per 5 minutes run. This is ~ 4.8 times more than the NC64 performance ($\sim 5,000$). We saw no 200-OK HTTP responses when

¹²<https://wiki.linuxfoundation.org/networking/netem>

the client was defenceless, i.e, the SYN flood attack traffic was 100% successful in breaching the availability of the victim. This shows the importance of SYN Cookies or NC64 defence during a SYN flood.

The evaluated performance of NC64 defence in LAN environments was expected for the following reasons.

1. MODCMS, CMS, and CMSVD daemons were not optimized for performance.
2. Communications among MODCMS, CMS, and CMSVD are secured by TLS whose layering on top of TCP connection introduces delays.
3. The network paths among DNS, CMS, and CMSVD introduce further latency for each capability request.

In lossy environments of 10% and 20%, we saw a similar performance of NC64 and SYN Cookies (under ~ 1000 200-OK packets per 5 minutes run). We chose these extreme lossy environments because enterprise networks might get extreme network congestion in the face of a DoS attack or TCB overflow throttling.

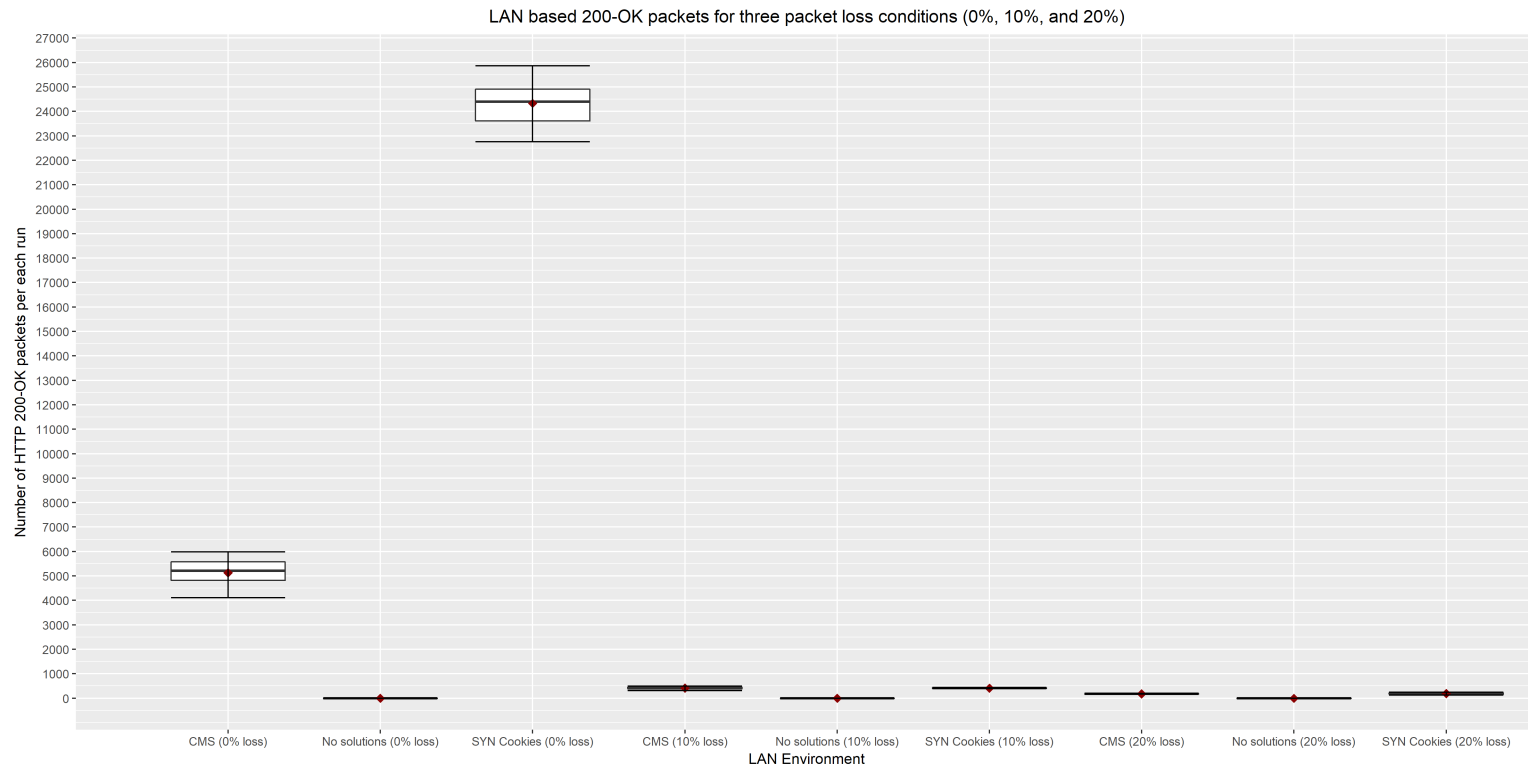


Figure 3.8: Performance evaluation of client-victim communication comparing lack of defence, SYN Cookies, and NC64 mechanisms. The diamond symbol in the figure shows the mean value.

3.3.2.2 MAN Environment

The Figure 3.9 shows results for a MAN environment with varying packet loss scenarios. It shows improvement in results for NC64 as compared to SYN Cookies LAN environment. SYN Cookies showed a throughput of $\sim 3,000$, and NC64 showed a throughput of $\sim 1,700$ in lossless conditions (~ 1.7 times better than NC64). If we compare it with lossy scenarios, NC64 is similar throughput as compared to SYN Cookies. In all lossy conditions in MAN, throughput is nearly zero for cases with defenceless victim. It should be noted that increasing the delay decreases the attack intensity but attack traffic relative to the client/server traffic has similar impact.

In terms of scientific and architectural aspects NC64 shows a successful proof of concept. In terms of engineering its performance can be optimized to give a better throughput.

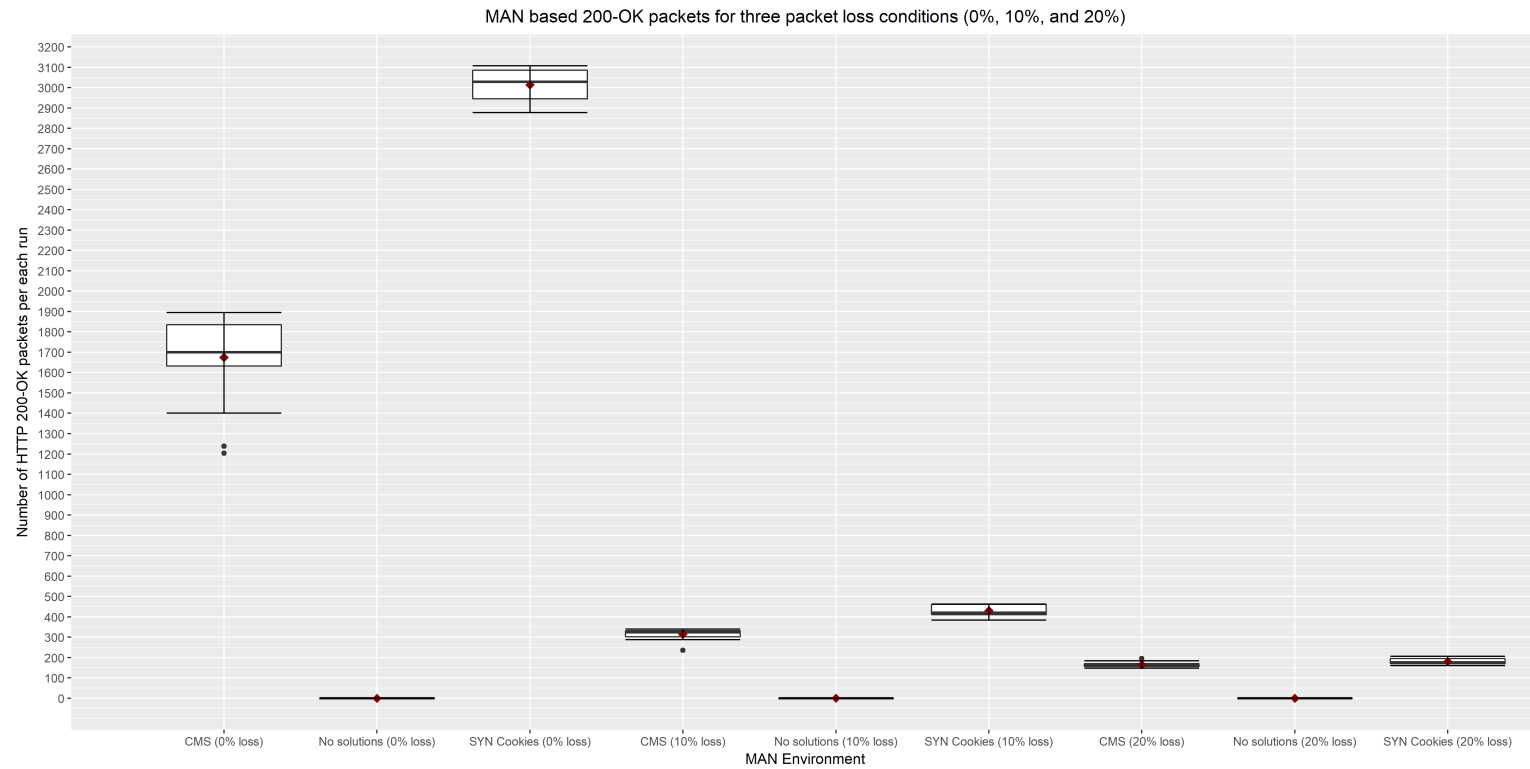


Figure 3.9: Performance evaluation of client-victim communication comparing lack of defence, SYN Cookies, and NC64 mechanisms. The diamond symbol shows the mean value.

3.3.2.3 WAN Environment

The Figure 3.10 shows the WAN based results. NC64 throughput is ~ 370 200-OK packets per 5 minutes run as compared to SYN Cookies which has ~ 390 . The throughput of SYN Cookies is ~ 1.054 times better than NC64 but if we consider error bars then NC64 performed similar to SYN Cookies.

There is a similar performance between the two in lossy conditions of 10% and 20%. We chose such extreme lossy conditions because an enterprise network might get extreme network congestion in the face of a DoS attack. Again, for defenceless victim, the client got less than ~ 10 200-OK packets, which shows the worst performance in 5 minutes duration.

It should be noted that increasing the delay decreases the attack intensity but attack traffic relative to the client/server traffic has similar impact, as has been noted in MAN environment.

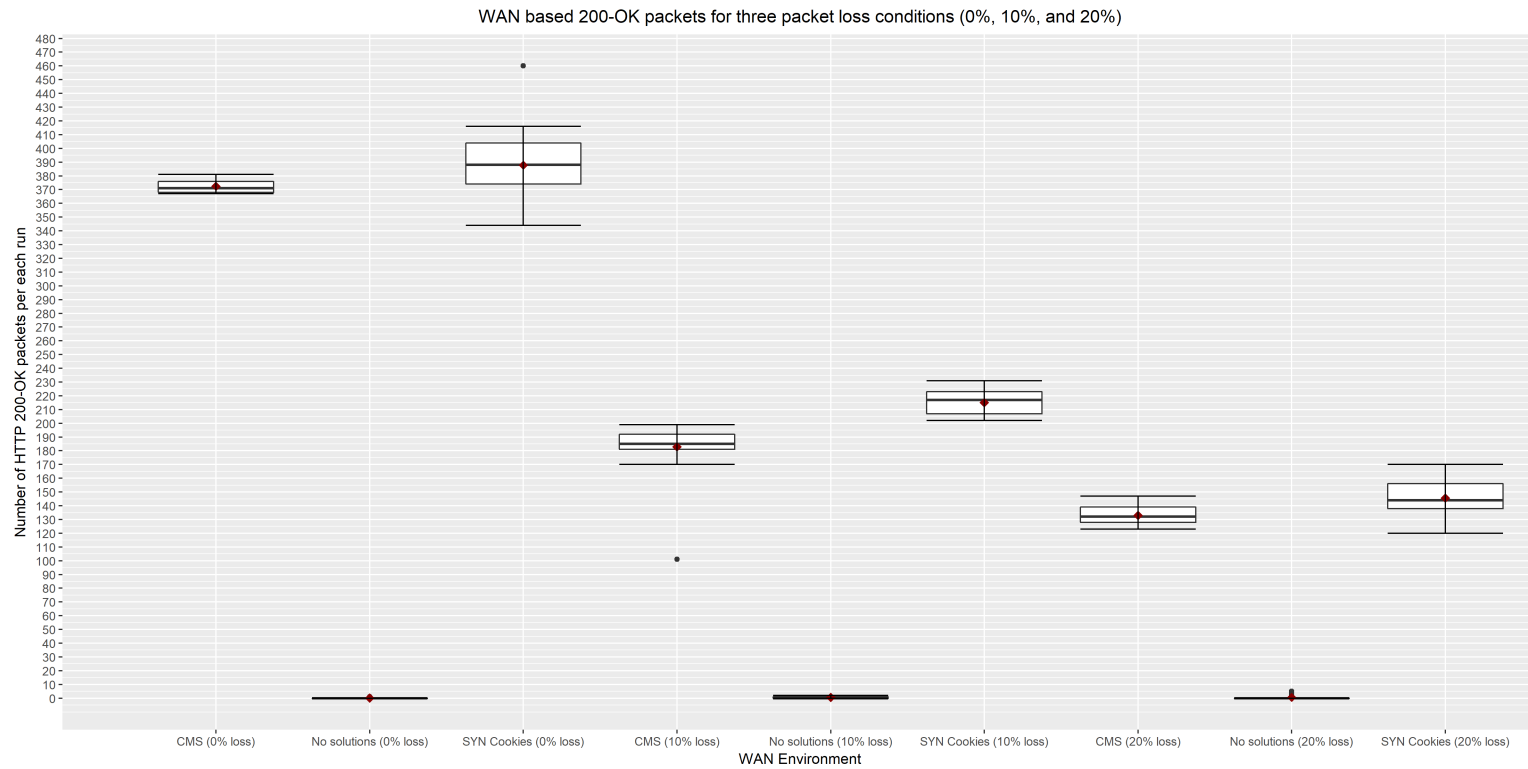


Figure 3.10: Performance evaluation of client-victim communication comparing lack of defence, SYN Cookies, and NC64 mechanisms. The diamond symbol shows the mean value.

3.3.3 Statistical Analysis

From the boxplots in Figures 3.8 to 3.10, we notice that statistical means and respective medians are similar (quantile-quantile plots show the same) which implies a normal distribution of the results. We are able to perform Welch's two-sample t-test significance testing on our results. Tables 3.1 to 3.3 show the statistical significance analysis of our null hypotheses (see §3.3.1).

We will reject the null hypothesis if NC64 defence did not perform similar to SYN Cookies ,i.e, if p-value is less than 0.05. If the p-value is equal to 0.05 then we can neither reject nor accept the null hypothesis. And, if the p-value is greater than 0.05 then we will accept the null hypothesis.

3.3.3.1 LAN Analysis

LAN Setup With No Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	<2.2e-16	<2.2e-16	<2.2e-16
Confidence Interval (99%)	18621.92 to 19772.08	4865.33 to 5400.19	23820.64 to 24838.88
Mean Values (ppr)	24329.76 and 5132.76	5132.76 and 0.00	24329.76 and 0.00
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG
LAN Setup With 10% Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	0.95	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-24.39 to 23.27	387.57 to 433.47	403.561 to 416.36
Mean Values (ppr)	409.96 and 410.52	410.52 and 0.00	409.96 and 0.00
Evidence For Acceptance (Similar True Means)	STRONG	STRONG	STRONG
LAN Setup With 20% Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	0.05 (or 0.05166)	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-4.97 to 33.85	169.84 to 181.68	171.71 to 208.69
Mean Values (ppr)	190.48 and 176.04	176.04 and 0.28	190.48 and 0.28
Evidence For Acceptance (Similar True Means)	NEITHER STRONG NOR WEAK	STRONG	STRONG

Table 3.1: Significance testing using Welch's two-sample t-test examinations of normally distributed results for three test cases within the LAN environment with varying packet loss conditions. Confidence intervals are at 99% level. *ppr* is packets per run.

The null hypothesis (see §3.3.1) can be rejected for no packet loss conditions. It can be accepted for 10% packet loss conditions. And it can neither be rejected nor accepted in 20% loss conditions. In lossy conditions, NC64

performs better because the traffic control daemons are able to serve the control traffic due to manageable capability request rate (a concurrent request handler for MODCMS can help further improve the request/response behaviour).

If we compare the NC64 performance with the performance of a defenceless victim, we see a strong evidence that the NC64 defence is better than having no defence (valid proof of concept). Similarly, if we compare the SYN Cookies performance with that of a defenceless victim, we see a strong evidence that the SYN Cookies defence is better than having no defence.

No Packet Loss Conditions:

The true statistical mean of the SYN Cookies performance is ~ 4.74 times more than NC64 for no packet loss conditions. Please see items 1 to 3 in §3.3.2.1 for reasons of such performance of NC64 defence which show that it is *an engineering issue rather than scientific*.

By comparing NC64 defence with a defenceless system, it can be said that the scientific nature of NC64 defence against SYN floods is established. Significant p-values of less than 0.05 and a large difference in respective means (i.e., means of the NC64 defence scenario and SYN Cookies only scenario) attribute to the aforementioned scientific contribution. So, an enterprise is encouraged to deploy NC64 defence if SYN Cookies is either not an option (see §§2.3.6.1 and 3.1) or if NC64 defence performance is increased using better engineering.

10% Packet Loss Conditions:

The true statistical mean of the SYN Cookies performance is ~ 1.0013 times more than NC64 for the 10% packet loss conditions, which implies that NC64 defence performs similar, on average, in lossy environments. But if we consider the p-value, then NC64 defence is better than SYN Cookies. The p-value of 0.95 is far more than 0.05 that implies that the SYN Cookies did not perform better in comparison to NC64.

NC64 defence performs better than having no SYN flood security in 10% packet loss conditions. The p-value is significantly less than 0.05, and the mean ratio is large, contributing to a better confidence interval. The same can be said about SYN Cookies and the defenceless system.

20% Packet Loss Conditions:

True statistical mean of the SYN Cookies performance is ~ 0.92 times more than the NC64 performance for the 20% packet loss conditions. In terms of mean values, NC64 performs better than SYN Cookies but if we consider p-value (~ 0.05) then there is no evidence that one is better than the other.

The NC64 defence performs better than having no SYN flood security in 20% packet loss conditions. The p-value is significantly less than 0.05, and the mean ratio is large contributing to a better confidence interval.

The network packet loss does reduce the magnitude of the attack traffic but the overall impact of the attack traffic has similar affect on the normal traffic, i.e., we still have attack SYN packets which overflow the TCP's TCB.

3.3.3.2 MAN Analysis

MAN Setup With No Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	<2.2e-16	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-1449.08 to -1228.44	1572.65 to 1777.75	2973.29 to 3054.63
Mean Values (ppr)	1675.20 and 3013.96	1675.20 and 0.00	3013.96 and 0.00
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG
MAN Setup With 10% Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	<2.2e-16	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-134.47 to -96.81	301.99 to 327.06	416.12 to 444.20
Mean Values (ppr)	314.52 and 430.16	314.52 and 0.00	430.16 and 0.00
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG
MAN Setup With 20% Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	0.00 (or 0.0001833)	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-25.32 to -5.16	158.08 to 171.04	172.08 to 187.52
Mean Values (ppr)	164.56 and 179.80	164.56 and 0.00	179.80 and 0.00
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG

Table 3.2: Significance testing using Welch's two-sample t-test examinations of normally distributed results for three test cases within the MAN environment with varying packet loss conditions. Confidence intervals are at 99% level. *ppr* is packets per run.

The null hypothesis (see §3.3.1) can be rejected for 0%, 10%, and 20% packet loss conditions in MAN environments. Even though the null hypothesis is rejected, we can say that NC64 defence is a valid proof of concept in emulated MAN environments, since they mitigated SYN attacks. Such performance of NC64 defence is expected due to latency issues by having extra network elements, and the unoptimized software implementations.

No Packet Loss Conditions:

True statistical mean of the SYN Cookies performance is ~ 1.80 times more than NC64 for no packet loss conditions, in a MAN environment. If compared with LAN, we have a reduction in ~ 3 times of the corresponding multiplication factor. According to the p-value, we can say that NC64 is not a better solution than SYN Cookies in terms of the performance of the 200-OK throughput metric. NC64 defence can increase its performance with better engineering.

By comparing the NC64 defence with the defenceless system, the NC64 defence performed better. The SYN Cookies also performed better than the defenceless system.

10% Packet Loss Conditions:

The true statistical mean of the SYN Cookies performance is ~ 1.37 times more than NC64 for 10% packet loss conditions. This shows that NC64 performs similar, in terms of means, to SYN Cookies in MAN based *lossy* environments. The difference in the two multiplication factors (~ 0.37), as compared to a LAN, signifies that packet losses in low data rate networks like MAN contribute more in degradation (although minimal) of NC64 defence rather than SYN Cookies. It is an odd observation for NC64 defence in MAN with 10% packet loss which demands further investigation into NC64 implementation details. If we compare it with a no-loss MAN environment, then it performs better. One reason for this is that during MAN-based delays and lossy conditions, CMS software is less loaded with capability requests than in no loss MAN or LAN environments.

NC64 defence performs better than having no SYN flood security in 10% packet loss conditions in MAN. The p-value is significantly less than 0.05, and the mean ratio is large contributing to a better confidence interval.

20% Packet Loss Conditions:

The true statistical mean of the SYN Cookies performance is ~ 1.09 times more than NC64 for 20% packet loss conditions. In terms of mean values NC64 performs similar to SYN Cookies but if we consider the p-value (~ 0.00) then SYN Cookies is better than NC64 defence.

NC64 defence performs much better than having no SYN flood security in 20% packet loss conditions in MAN. The p-value is significantly less than 0.05, and the mean ratio is large, contributing to a better confidence interval. The same can be said about the comparison of the SYN Cookies with the defenceless system.

3.3.3.3 WAN Analysis

WAN Setup With No Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	0.00 (or 0.005458)	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-29.21 to -1.19	369.80 to 374.85	373.74 to 401.30
Mean Values (ppr)	372.52 and 387.72	372.52 and 0.20	387.72 and 0.20
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG
WAN Setup With 10% Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	4.36e-10	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-43.32 to -21.16	172.24 to 192.32	209.78 to 219.25
Mean Values (ppr)	182.88 and 215.12	182.88 and 0.60	215.12 and 0.60
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG
WAN Setup With 20% Packet Loss	NC64 vs SYN Cookies	NC64 vs No Solutions	SYN Cookies vs No Solutions
p-value	6.39e-05	<2.2e-16	<2.2e-16
Confidence Interval (99%)	-20.51 to -4.93	128.40 to 136.08	138.08 to 151.83
Mean Values (ppr)	132.96 and 145.68	132.96 and 0.72	145.68 and 0.72
Evidence For Acceptance (Similar True Means)	WEAK	STRONG	STRONG

Table 3.3: Significance testing using Welch’s two-sample t-test examinations of normally distributed results for three test cases within the WAN environment with varying packet loss conditions. Confidence intervals are at 99% level. *ppr* is packets per run.

The null hypothesis (see §3.3.1) can be rejected for 0%, 10%, and 20% packet loss conditions. Even though the null hypothesis is rejected, we can say that NC64 defence is a valid proof of concept in a WAN environment as well (as is the case in LAN and MAN), as it mitigated the SYN flood attack.

No Packet Loss Conditions:

The true statistical mean of SYN Cookies performance is ~ 1.04 times more than NC64 for no packet loss conditions in WAN environment. If compared with the LAN based analysis, we have a reduction of ~ 4.8 times of the corresponding multiplication factor. In terms of the p-value, we can say that NC64 is not a better defence than SYN Cookies. The reduction in the multiplication factor is a better result, but the p-value clearly shows that the NC64 implementation requires better engineering and low backend latency.

NC64 did perform similar to SYN Cookies if we look at the width of the confidence interval, and the difference in means.

The NC64 defence and the SYN Cookies defence performed better than the defenceless system.

10% Packet Loss Conditions:

The true statistical mean of the SYN Cookies performance is ~ 1.17 times more than NC64 for 10% packet loss conditions. The difference in two multiplication factors, as compared to LAN, is minimal in 10% packet loss case, meaning it performed similar to LAN in such conditions.

NC64 defence performs better than having no SYN flood security in 10% packet loss conditions in WAN. The p-value is significantly less than 0.05, and the mean ratio is large contributing to a better confidence interval. The same can be said about the SYN Cookies versus no defence.

20% Packet Loss Conditions:

The true statistical mean of the SYN Cookies performance is ~ 1.10 times more than the NC64 performance for the 20% packet loss conditions in the WAN environment as well in comparison to the MAN environment. Even though from the outset it looks like a similar result, if we consider the p-value ($\sim 6.39\text{e-}05$), we need to say that the alternative hypothesis of our null hypothesis (see §3.3.1) will take preference.

The NC64 defence performs better than having no SYN flood security in the 20% packet loss conditions in the WAN environment as well. The p-value is significantly less than 0.05, and the mean ratio is large contributing to a better confidence interval. The same can be said about the SYN Cookies only versus no defence.

3.3.4 NC64 And SYN Cookies — Performance Similarity And Overall Benefits

Given tables 3.1 to 3.3, we notice that the overall performance of NC64 is not better than SYN Cookies. NC64 performance can be increased through better engineering (which is a future work). Poor NC64 performance can be reconciled by using its other benefits which can not be obtained from SYN Cookies.

NC64 supports data piggybacking on initial TCP packets, which are not supported by SYN Cookies. NC64 supports TCP options, but SYN Cookies are not compatible with TCP options. NC64 have no problems with unidirectional data flow when Selective ACK (SACK) packets are in use, whereas SYN Cookies have problems with them. Please see §§2.3.6.1 and 3.1 for problems associated with SYN Cookies. As NC64 does not utilize mechanisms associated with modifications of TCP protocol stack and

TCP implementation within the Linux kernel, it does not suffer from such shortcomings.

3.4 Evaluating Performance Of NC64 Distribution

In NC64, the capability distribution mechanism requires having a CMS back-end. It is important to see performance of the CMS backed DNS responses and compare it with the performance of DNS only responses to clients. For such purposes we ran an experiment with Local Area Network (LAN), Metropolitan Area Network (MAN) (emulated), and Wide Area Network (WAN) (emulated) environments.

3.4.1 Experiment Design

Figure 3.11 shows a logical diagram of this experiment with a reference enterprise network. The CMS is contained in the enterprise's protected network behind the Demilitarized Zone (DMZ). As the DMZ is the entry point of any external traffic, we have put DNS and the publicly accessible victim in it. The DNS and the CMS are connected through an internal router and there is a link between the CMS and the victim to derive firewall rules. We have emulated different network conditions between the client and the enterprise network.

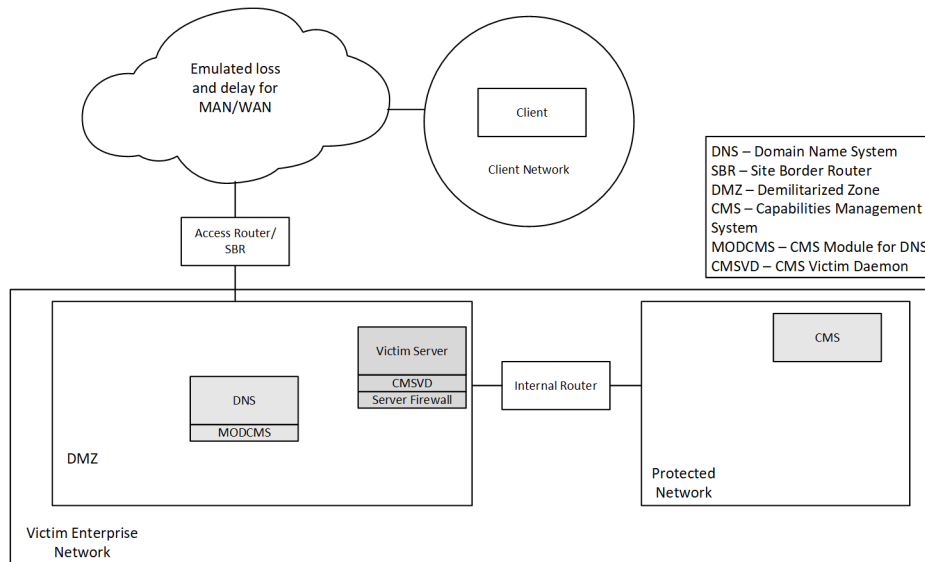


Figure 3.11: Logical diagram for the performance comparison of NC64 capability-backed DNS response distribution and DNS only response distribution to clients

The experiment has two parts. The first part used a DNS response

mechanism without CMS, and second part did the same experiment with CMS in place. There were 25 iterations done for each part, and performance measurements were taken using average response times. Each iteration contained 100 DNS requests. After each iteration, the average response time was calculated and stored as a single value which is representative of the respective iteration.

3.4.2 Testing

The main requirement for testing was to generate client requests and to collect response times for each response for DNS with and without CMS scenarios. The client used the `dig`¹³ software, which is natively available in Linux distributions, to launch a series of DNS requests. A custom script was created to generate 100 requests per iteration for each scenario. A monitoring agent was running at the client to collect actual response times. Emulated packet loss of 0%, 5%, and 10% was used in the LAN, MAN, and WAN environments, respectively, in each direction between the client and the DNS. We chose lossy environments of 5% and 10% in each direction to measure performance when there is extreme network congestion during a DoS attack. We get an aggregate of 10% and 20% packet loss between client and DNS.

Similarly, we introduced an emulated delay of 12.5ms, and 105ms in each direction between the client and the DNS for the MAN and the WAN cases, respectively. So, in aggregate, we get a delay of 25ms and 210ms between the client and the DNS for the MAN and the WAN environments. For the LAN case, we use the usual network conditions in the testbed without network emulation. We chose these delay figures after doing some repeated measurements of actual MAN, and WAN environments, e.g., using `dig` to measure the response time of University's services and geographically distributed public servers. We picked the average values and used them in the experiment for emulation.

We used `netem`¹⁴ software, which is widely accepted as a network emulation software for the generation of loss and delay.

All systems in the testbed run Ubuntu server 16.04. The client is running an ILNP kernel version 4.9.38. The DNS and the CMS run on an IPv6 kernel version 4.9.38 because there is no requirement on being used as an ILNP machine.

Each router is directly connected to an Extreme[®] Switch x45a-48t with an isolated port, using 1 Gbps full-duplex Ethernet links. Each machine, apart from the switch in the testbed is a Gateway[®] GR380 F1 machine with 64-bit Intel[®] Xeon[®] 8-core CPU (2.27 GHz base frequency)¹⁵.

¹³[ftp://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/man.dig.html](http://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/man.dig.html)

¹⁴<http://manpages.ubuntu.com/manpages/xenial/man8/tc-netem.8.html>

¹⁵https://ark.intel.com/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-

3.4.3 Results

LAN Environment

Figure 3.12 shows the LAN-based results.

When we introduced no packet loss in the LAN, we saw a performance of less than 50 milliseconds for each scenario. For the DNS only scenario, we saw ~ 16 ms of average response time, and for the DNS with a CMS scenario we saw ~ 42 ms of average response time. The DNS without CMS case performed ~ 2.625 times better than the DNS with CMS. ~ 42 ms of average response time for DNS with CMS is not good enough but the results give us information on how much better we need to do.

In the case of 10% packet loss, we saw an average response time of ~ 532 ms for the DNS only scenario, and ~ 558 ms for the DNS with CMS scenario. The DNS without CMS performed ~ 1.049 times better than the DNS with CMS.

Similarly, in the case of 20% packet loss, we saw an average response time of ~ 1194 ms for the DNS only scenario and ~ 1116 ms for the DNS with CMS scenario. The DNS with CMS performed ~ 1.070 times better than the DNS without CMS.

So, in lossy conditions, the DNS with CMS has similar performance with the DNS without CMS. Whereas in lossless conditions, the DNS without CMS is better than the DNS with CMS. The difference is due to an unoptimized backend, use of TLS for all backend control signalling, and having an extra latency due to new backend network components. Some of these parameters can be controlled, e.g., if the CMS is behind the DMZ then the enterprise can relax the TLS dependency. Similarly, if the participating softwares, i.e., CMS, CMSVD and MODCMS, are designed and implemented in such a way so as to deliver high performance (e.g. using concurrency or parallelism), then we would be able to enhance the performance of the scenario for DNS with CMS.

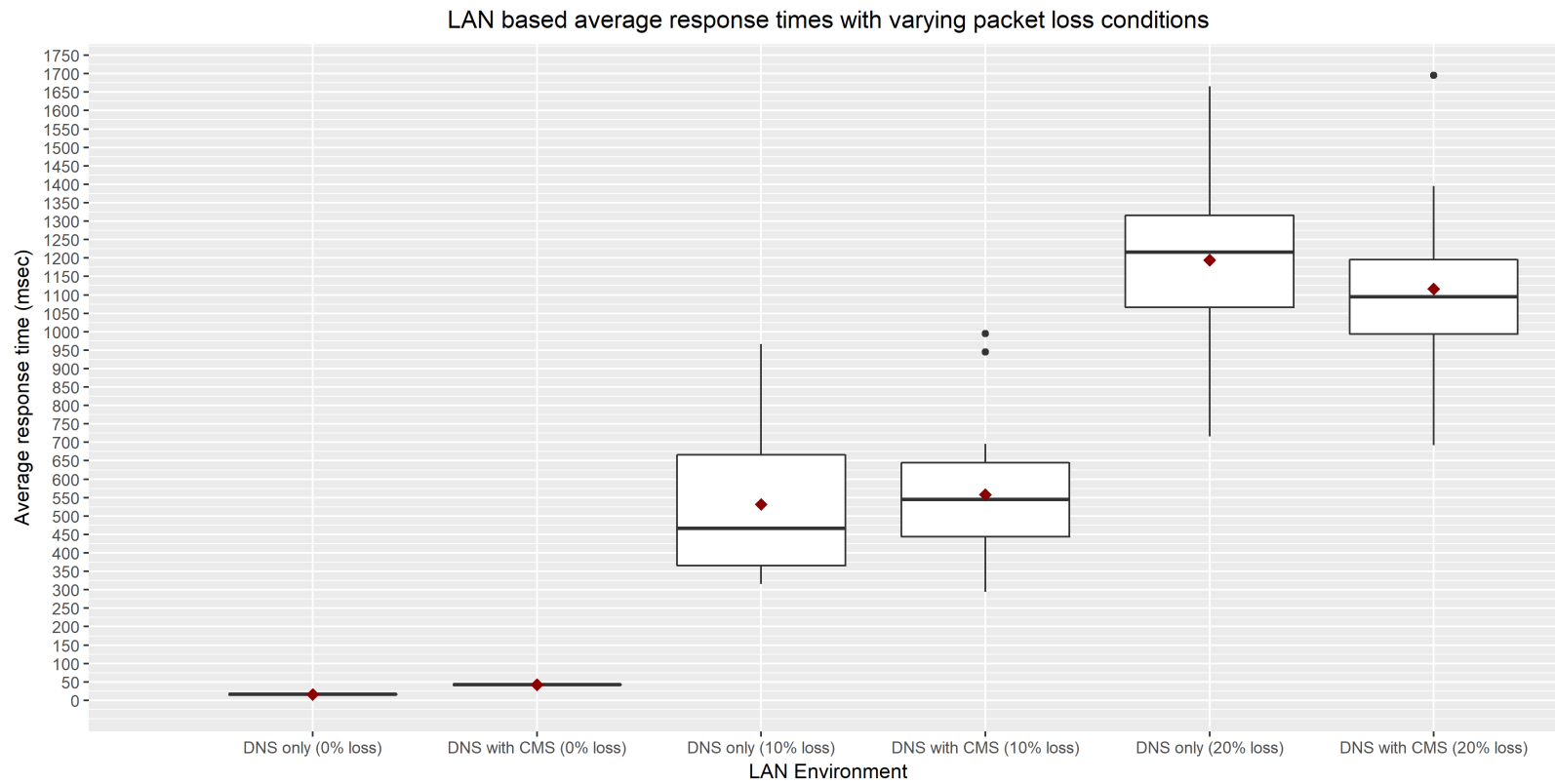


Figure 3.12: Average response time measurements (milliseconds) taken for the LAN environment with a 0%, 10%, and 20% emulated packet loss. The graph shows information about the scenarios of measuring DNS requests/responses with and without CMS

MAN Environment

Figure 3.13 shows MAN based results.

When we introduced no packet loss in the MAN, we saw a performance of less than 100 milliseconds for each scenario. For the DNS only scenario for the MAN, we saw ~ 42 ms of average response time, and for the DNS with CMS scenario, we saw ~ 67 ms of an average response time when there was no packet loss. The DNS without CMS performed ~ 1.595 times better than the DNS with CMS.

In the case of 10% packet loss, we saw an average response time of ~ 551 ms for the DNS only scenario, and ~ 603 ms for the DNS with CMS scenario. The DNS without CMS performed ~ 1.094 times better than the DNS with CMS.

Similarly, in the case of 20% packet loss, we saw an average response time of ~ 1209 ms for the DNS only scenario and ~ 1201 ms for the DNS with CMS scenario. The DNS with CMS performed ~ 1.007 times better than the DNS without CMS.

So, in the lossless and lossy conditions, the DNS with CMS has similar performance to the DNS without CMS. The difference is due to an un-optimized backend, use of TLS for all the backend control signalling, and having additional latency due to new backend network components. The improvement in the MAN environment comparison to the LAN environment is due to network delays which allow the CMS software to perform better while calculating/selecting the NC64 values and consuming/producing the control messages from/to other systems.

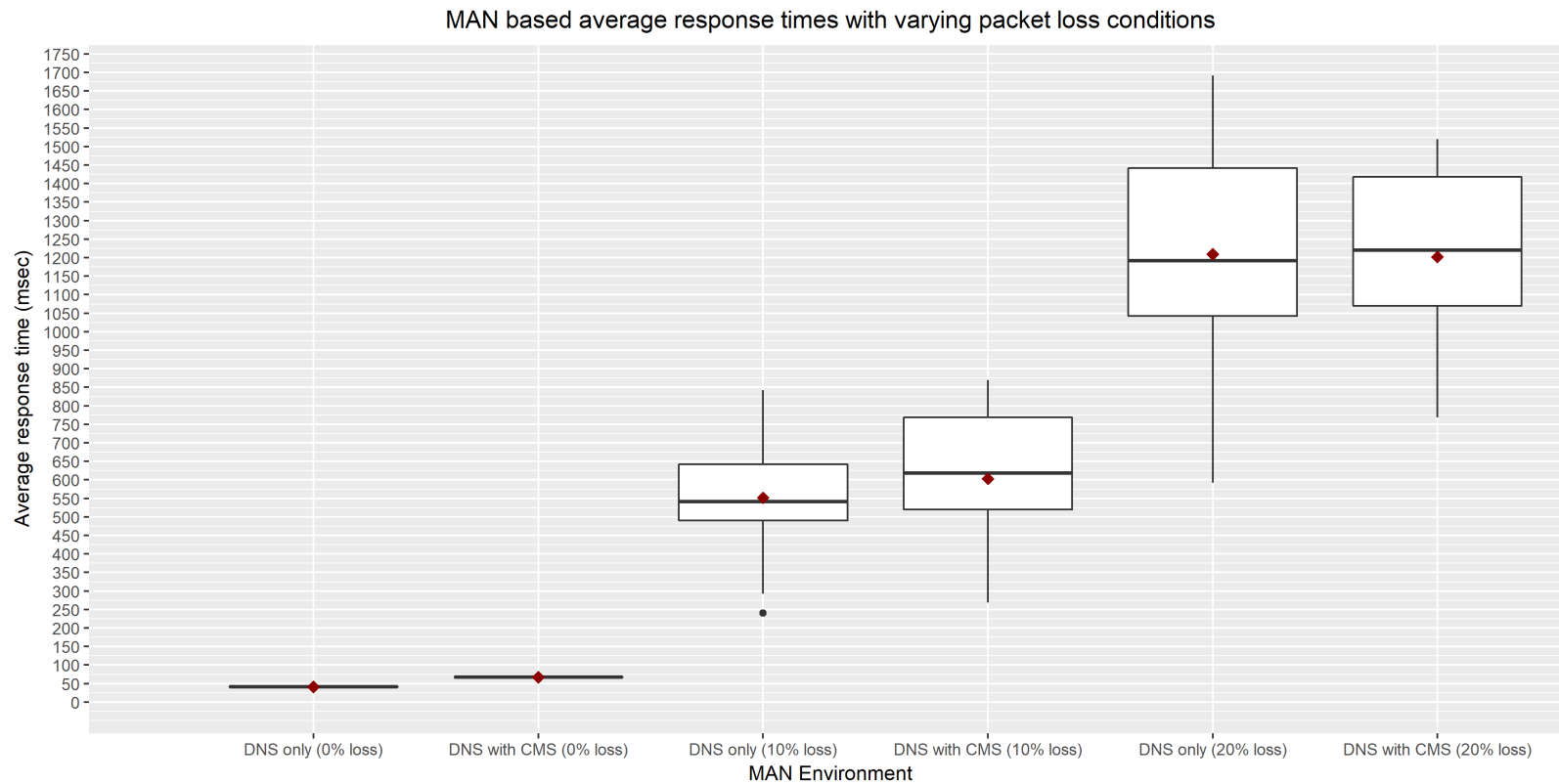


Figure 3.13: Average response time measurements (milliseconds) taken for the MAN environment with a 0%, 10%, and 20% emulated packet loss. The graph shows information about the scenario of measuring DNS requests/responses with and without CMS

WAN Environment

Figure 3.14 shows WAN based results.

When we introduced no packet loss in the WAN environment, we saw a performance of less than 250 milliseconds for each scenario. For the DNS only scenario for the WAN, we saw ~ 226 ms of an average response time, and for the DNS with CMS scenario, we saw ~ 254 ms of an average response time when there was no packet loss. The DNS without CMS performed ~ 1.124 times better than the DNS with CMS.

In the case of 10% packet loss, we saw an average response time of ~ 792 ms for the DNS only scenario, and ~ 810 ms for the DNS with CMS scenario. The DNS without CMS performed ~ 1.023 times better than the DNS with CMS.

Similarly, in the case of 20% packet loss, we saw an average response time of ~ 1454 ms for the DNS only scenario and ~ 1409 ms for the DNS with CMS scenario. The DNS with CMS performed ~ 1.032 times better than the DNS without CMS.

So, in lossless and lossy conditions, the DNS with CMS has a similar performance in comparison with the DNS without CMS. The improvement in the WAN case in to the LAN is due to the network delays which allow the CMS software to perform better while calculating/selecting the NC64 values and consuming/producing the control messages from/to other systems.

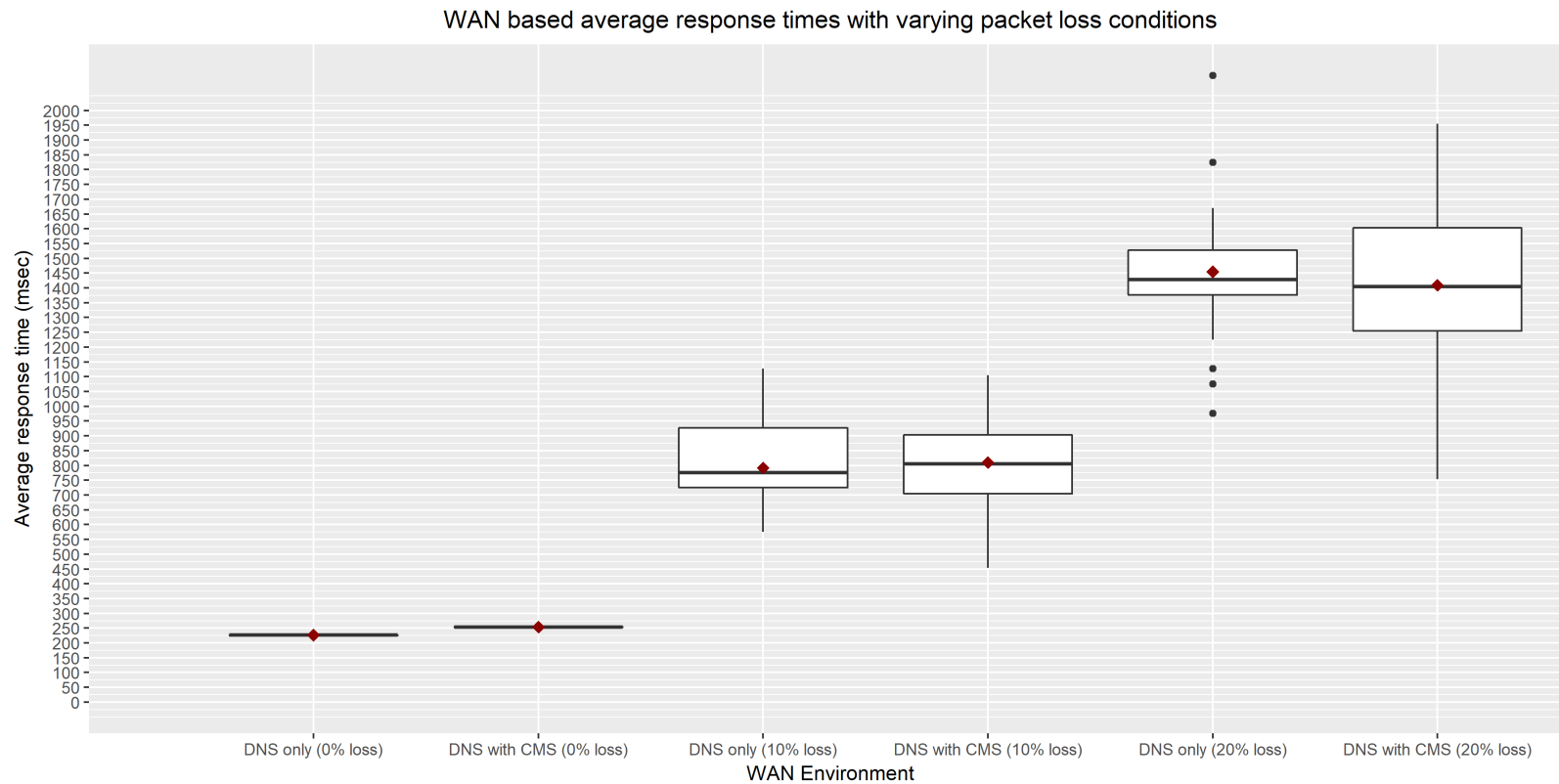


Figure 3.14: Average response time measurements (milliseconds) taken for the WAN environment with a 0%, 10%, and 20% emulated packet loss. The graph shows information about the scenario of measuring DNS requests/replies with and without CMS

Table 3.4 summarizes the measurements reported in this section.

LAN Setup with Packet loss	Average Response Time (DNS only)	Average Response Time(DNS with CMS)
0%	~16ms	~42ms
10%	~532ms	~558ms
20%	~1194ms	~1116ms
MAN Setup with Packet loss	Average Response Time (DNS only)	Average Response Time(DNS with CMS)
0%	~42ms	~67ms
10%	~551ms	~603ms
20%	~1209ms	~1201ms
WAN Setup with Packet loss	Average Response Time (DNS only)	Average Response Time(DNS with CMS)
0%	~226ms	~254ms
10%	~792ms	~810ms
20%	~1454ms	~1409ms

Table 3.4: Summary of response time measurements taken for DNS-only and DNS with CMS scenarios under different packet loss conditions

3.5 Summary

This chapter covered the ILNP identifier namespace-based DoS defence system for an enterprise host security. It covered the rationale for the research (see §3.1), implementation details (see §§3.2.1 and 3.2.6), the testing details (see §3.3) and the results along with their statistical analysis (see §§3.3.2 and 3.3.3). We termed this defence as NID64-based capabilities (NC64). These capabilities provide per-client authorization to allow a client to access services running on enterprise hosts.

Mitigation against low-rate TCP SYN flood attacks is the main objective of this defence. An evidence of ILNP-based defence feasibility and proof of concept was supported empirically by comparing it with a widely deployed non-ILNP defence, i.e., TCP SYN Cookies. The ILNP-based defence not only provides security against TCP SYN flood attacks but it also curbs the problems that arise when TCP SYN Cookies are in use (see §3.3.4).

This chapter also presented an empirical evaluation of the performance of NC64 capability distribution to clients (see §3.4). We noticed that the latency introduced in the distribution mechanism is mainly due to the unoptimised implementation of the new software elements, and the extra network infrastructure required for the control traffic for the defence (see §3.4.3).

Chapter 4

Protecting Sites from Spoofing based Volumetric DoS Attacks

Enterprises put multi-level anti-DoS (see §2.3.7) security controls in their systems and networks. In this chapter, we will evaluate a defence mechanism based on ILNP’s 64-bit locator namespace (L64) which is used as an IPv6 network prefix. L64-based defence is expected (using ILNP mobility and MTD) to protect an enterprise site from volumetric DoS attacks.

This chapter presents the following parts:

- Establishing the rationale for investigating L64-based defence against volumetric DoS attacks. We cover shortcomings of current mitigations against these assaults. Definitions of new concepts, and relevant conditions/assumptions in their applicability will also be presented.
- Empirically comparing the L64-based defence mechanism to the case of an unprotected site during an attack. Our baseline will be a defenceless client-server communication. The effects of volumetric UDP floods on client-server communication will be measured by comparing client traffic during an attack, with and without our solution.
- Empirically evaluating the effects of L64-based defence on (volumetric) SYN flood attacks. Here, our objective is to quantify how much bandwidth coverage an attacker loses when our solution is in place. *We did not use client traffic in this evaluation, i.e., to assume that attack is 100 percent successful before applying the solution.* Our aim is to mitigate DoS attack by increasing the effort required by the attacker.
- Measuring SYN flood attack traffic when an L64-based defence uses network paths which include the path on which the attack is occurring.

4.1 Rationale

Chapter 3 evaluated the NID64-based capabilities (NC64) that protected enterprise servers from DoS. If an enterprise, *only*, deploys a NC64 defence, then volumetric attack traffic will still be able to reach an enterprise site because the NC64 defence uses different NID64 values at the victim host while the topological path between it and the external network entities remains the same. So, in NC64 defence, an enterprise network is able to see all network packets but the victim host will not be able see the packets from unauthorized sources. An enterprise requires a different defence against unauthorized packets which can reach its network.

We propose a mechanism based on ILNP's L64 namespace. It uses the same methodology of MTD (see §1.4), capabilities (see §1.5), and DNS Fast Flux (see §1.6) as has been used by the NC64 defence. We term this solution as *L64-based Capabilities* (LC64).

In ILNP, NID64 and L64 values serve two different purposes (see §§2.1.1.1 and 2.1.1.2). Each namespace can be used *dynamically* (see §2.1.1) because they are separately used in the transport and network layers. ILNP provides mechanisms to control dynamic bindings (see §2.1.2.1). The NID64 values can dynamically bind with any network interface during the same communication session. NID64 values change their network/interface bindings using Locator Updates (LUs) (see §2.1.3). We will use this dynamic binding mechanism and the LU mechanism to form the LC64 defence.

In ILNP, continuity of a communication session is dependent on NID64 invariance but is independent from L64 variance. So, we can change L64 values (or topologically change network access) while maintaining existing sessions. In relation to the MTD paradigm, we can use L64 values as a *moving* property (see §2.4.1.1) with network access authorization enabled through the capabilities paradigm.

In our solution, dynamically changing network topology requires multiple upstream links (with unique paths) to the Internet. In case of IPv6, if the target changes its location, then all TCP connections have to be broken and re-established. But in the case of ILNP, existing TCP connections are maintained when the victim changes its topological location [Atkinson & Bhatti, 2012a]. We make use of these ILNP features which enable host/network mobility without disrupting end-to-end communications.

For a description of ILNP mobility please see §2.1.3. It should be noted that ILNP mobility has been built into the ILNP kernel (version 3.9.x and 4.9.38) and evaluated empirically in earlier research [Phoomikiattisak, 2016].

4.1.1 Definitions And Assumptions

4.1.1.1 ILNP Locator-based Capabilities (LC64)

In LC64 defence, each ILNP locator (L) value is an ephemeral L64 namespace (MTD compliant) that gives network access to an authorized client for a short duration. There must be more than one LC64 value, and it is a requirement to frequently change or rotate them within the same communication session hence achieving a topologically disparate multi-path environment.

4.1.1.2 NC64 vs LC64

In §3.2, NID64 based Capabilities (NC64) were defined along with a required MAP-based filtering at the victim server. But, LC64 does not use filtering, hence no mappings. While IPv6 can implement IP-based capabilities, similar to NC64, defence similar to LC64 is not possible with IPv6 unless mobility extensions for IPv6 are in use. *LC64 is purely an ILNP solution and it requires that client and server must both support ILNP.*

LC64 defence can only divert traffic, but if any traffic is leaked to the network then it will reach applications running on the victim server. In NC64 defence, traffic has to acquire a LC64 capability which has a state in the victim server's local firewall, hence services are protected from unauthorized sources.

Both NC64 and LC64 defences are solutions against spoofing-based DoS attacks. NC64 defence can work with low data-rate and high data-rate SYN floods but it is better suited for low-rate SYN floods. LC64 is intended to work against volumetric i.e. high data-rate DoS attacks which can be either UDP or TCP SYN floods. Here the notion of a flood is anything that overwhelms a certain resource in the system, e.g., SYN flood (even though it mostly comes with a low data rate) overwhelms the TCP's TCB buffers.

LC64 defence requires a Capabilities Management System (CMS) as an extra network element which has different functionality than the NC64 CMS. What follows is a description of the extra network elements required for an LC64 defence.

4.1.1.3 LC64 MODCMS: LC64 CMS Module Within DNS

LC64 MODCMS is a knotDNS¹ module that is designed and built using C by myself, under the supervision and guidance of my supervisor. It has the same functionality as an NC64 MODCMS.

In this chapter, we will use the term MODCMS in the context of an LC64 defence. Its responsibility is to extract client information from a DNS name resolution request; and to create a secure LC64 capability request to be sent to the LC64 CMS. Once it has received an LC64 response from the

¹<https://www.knot-dns.cz/>

LC64 CMS, it strips this information off and returns a valid DNS name resolution response to the client via DNS. For a complete description of the MODCMS please see §§3.2.1.2 and 3.2.5.2.

4.1.1.4 *LC64 Capabilities Management System (LC64 CMS)*

LC64 Capabilities Management System (LC64 CMS) is an enterprise protected network entity that controls LC64 creation, compliance, and temporal allocation. It is connected to each enterprise-owned upstream access router that provides a different network path through the Internet. An LC64 CMS controls the interfaces on each access router. It can also configure paths in the routing tables of these routers.

CMS, as an extra network element, is common to NC64 and LC64 defences, but each CMS instance has a different responsibility. In NC64, a CMS controls mappings in the victim server, whereas in LC64, it only controls router interfaces/paths. An LC64 CMS does not interact with a victim host.

4.1.1.5 *Assumptions*

In LC64 defence, we assume that

- an attacker is using IP spoofing to launch a UDP or a TCP SYN flood.
- an enterprise is connected to at-least two upstream links.
- LC64 values are changed in 20 seconds intervals. Since we are not measuring network hand-off performance, we can choose any interval as long as we have multiple changes in topological paths within a single observable communication flow (to create a proof of concept).
- Location and identity of an LC64 CMS is only known to an enterprise.
- An LC64 CMS is protected by a DMZ, whereas a DNS is located within a DMZ.
- Client and enterprise machines uses an ILNP kernel.

4.2 Empirical Evaluation Of LC64 Defence

4.2.1 Defence Protocol

The defence follows following steps in order:

1. An external correspondent node (CN), i.e., a client, sends a DNS name resolution request to a DNS.

2. The DNS sends capability request to an LC64 CMS.
3. The LC64 CMS picks up a currently active L64 value from a pool of L64 values, and enables routing on the active link. It also, disables the routing on all the inactive L64 uplinks.
4. The LC64 CMS sends the capability reply to the DNS.
5. The DNS sends the DNS name resolution response message to the CN by embedding the new LC64 value.
6. The client can now connect to the victim using only this active capability, with a newly configured path.
7. During the client server data communication, the CMS shifts to other locator values after some configurable interval while the ILNP protocol maintains the session (see §2.1) .

Once the LC64 defence protocol is active, spoofed attack traffic will not reach the enterprise network, unless the interface on which the attack is launched is active as well.

Figure 4.1 shows an end-to-end sequence diagram for frequent L64 namespace reassignment using the LC64 defence. It showcases two LC64 namespaces controlled by an enterprise LC64 CMS, and advertised by two routers to an internal enterprise server. This server then uses ILNP mobility to shift existing connections to a new topological path.

If a spoofing based attack traverses through router one, then legitimate clients see an increase in bandwidth as soon as the LC64 defence shifts to a different topology using router two. Each legitimate client is shifted to the new topology using ILNP-based mobility mechanism which contains control signalling that only legitimate clients can acknowledge. For a successful DoS, an attacker has to increase its effort (see §4.3) by attacking on all available upstream links of the enterprise.

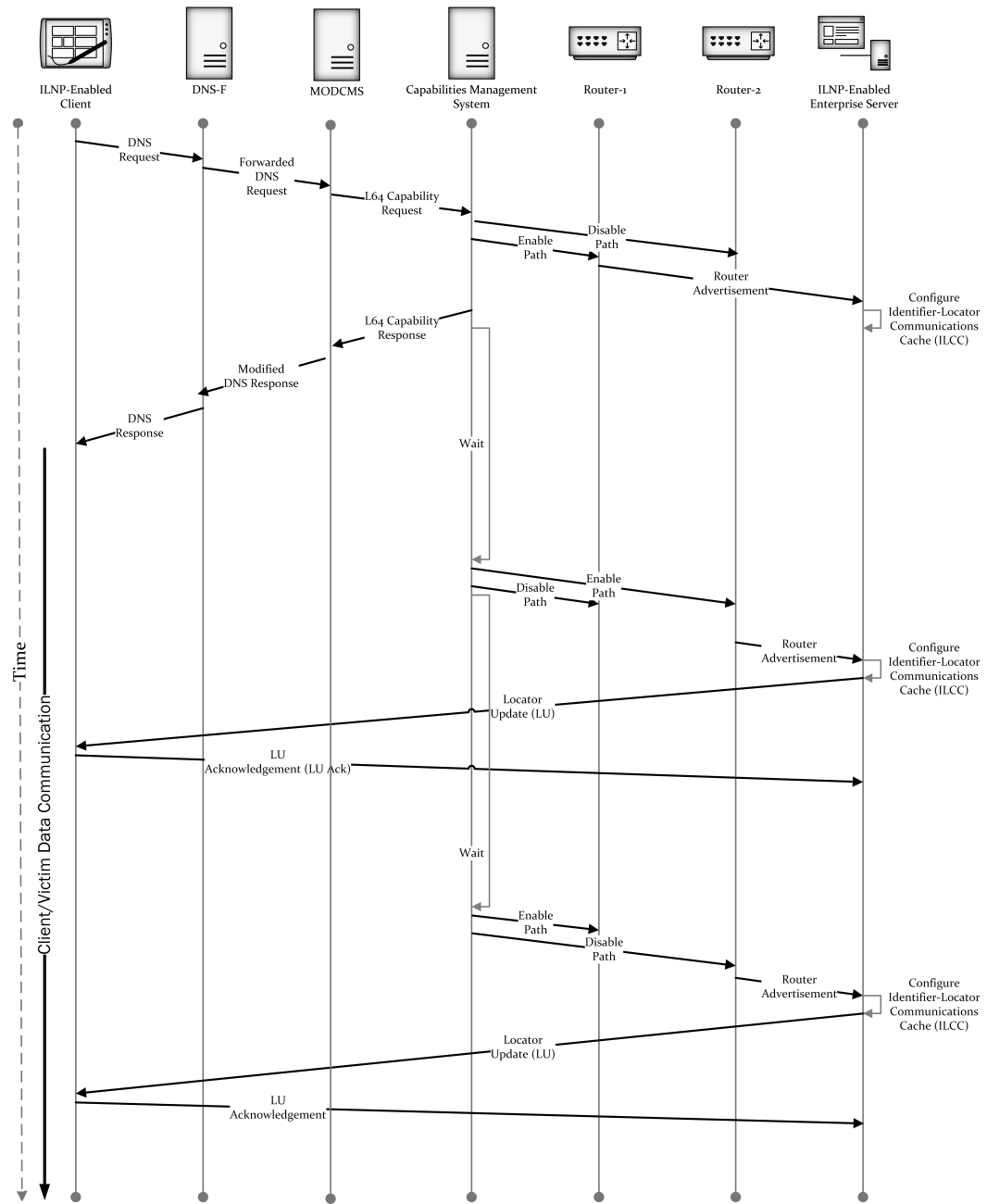


Figure 4.1: LC64 Defence Protocol Sequence Diagram

4.2.2 Enterprise Network Architecture With New Network Components

Figure 4.2 shows the logical diagram of an example enterprise network, connected to the Internet, with a proposed placement of DNS and LC64 CMS.

The internetwork contains a client network, and a victim network that contains DNS and publicly accessible servers in a DMZ along with LC64 CMS in the protected network (i.e., behind DMZ). Two upstream links are available through two access routers. The enterprise site is designed to maximize the security of an LC64 CMS by putting it behind the DMZ. It is recommended to have firewalls in the DMZ and the protected networks. The MODCMS is shown as a collocated component within the DNS server. It should be noted that there is no CMSVD as it was only required in the NC64 defence (see chapter 3).

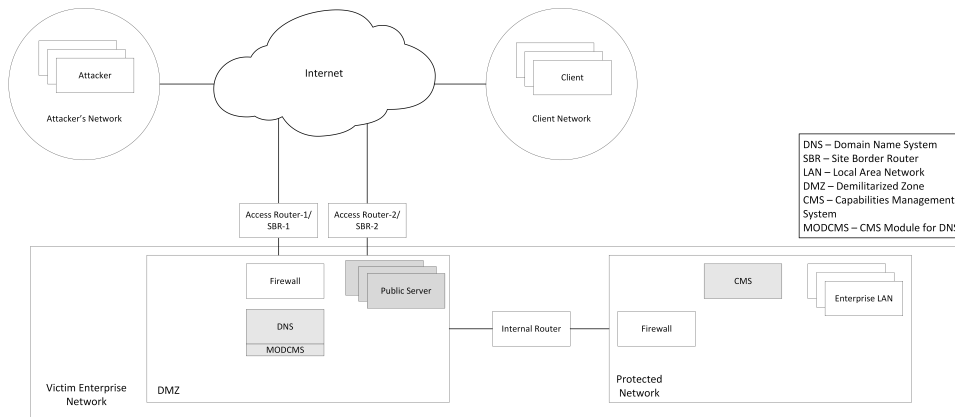


Figure 4.2: The reference inter-networked enterprise site for which an LC64 defence is designed

Domain Name System (DNS)

An LC64 defence requires DNS server software. A client can use TCP, UDP, DNSSEC or any other protocol that is supported by DNS server software. An LC64 defence requires an interface (MODCMS §4.1.1.3) to DNS that would generate L64 capability requests to the LC64 CMS and consuming L64 capability responses from the LC64 CMS. This interface is between an LC64 CMS and a DNS (MODCMS component within DNS) and it does not communicate with a client.

LC64 MODCMS (LC64 CMS Module For DNS)

A LC64 MODCMS has the same functional architecture as a NC64 MODCMS. For details, please see §4.1.1.3. Figure 4.3 shows the FSM of MODCMS. We reproduce its FSM diagram here for completeness.

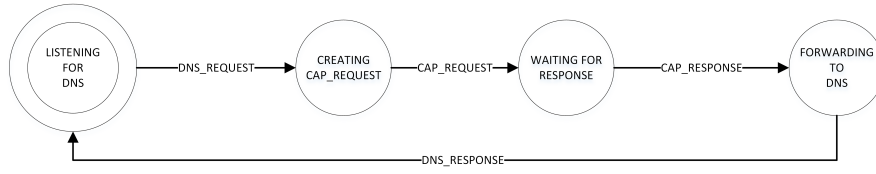


Figure 4.3: MODCMS Finite state machine

LC64 CMS

LC64 CMS component is used to control the uplinks associated with an enterprise. It controls which uplinks should be active or inactive at any specific time.

A LC64 CMS has different functionality than an NC64 CMS. It maintains a list of route-able site-specific L64 namespaces whereas an NC64 CMS maintained a list of host-specific NID64 values. An LC64 CMS connects to all upstream routers and DNS through a secure (SSH in our case) protocol. Each L64 namespace that it makes active (using dynamic router configurations) becomes a capability for a short duration. This lifetime can be controlled by enterprise management systems. The main responsibilities of the LC64 CMS are:

- Selecting an active ephemeral LC64 value
- Activating paths on access routers with the active ephemeral LC64 value
- Deactivating paths on access routers with in-active LC64 values
- Responding to a MODCMS DNS module with an active LC64 value once it receives an LC64 capability request from a MODCMS.

Figure 4.4 shows the finite state machine of an LC64 CMS.

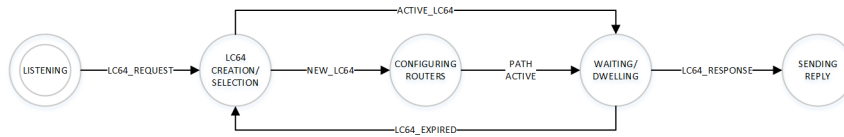


Figure 4.4: LC64 CMS Finite state machine

Where

- A LISTENING is a state while a LC64 CMS is listening for LC64 requests from a MODCMS.

- An LC64 CREATION/SELECTION is a state where a LC64 CMS either creates a new ephemeral LC64 value based on the time-out associated with the currently active LC64 value or picks a value which is not yet timed out.
- A CONFIGURING ROUTERS is a state which is executed after a new LC64 value is selected. In this phase, it performs two parallel tasks. The first task runs a script on the active router to update the radvd daemon configuration and configures the ingress path on the router's network interface using iptables². The second parallel task runs a script on a router whose L64 value is now inactive, to deactivate the ingress path using iptables.
- In the WAITING/DWELLING state, the LC64 CMS waits for a time-out after which the current ephemeral LC64 value becomes inactive.
- Once LC64 CMS is in a WAITING/DWELLING state, it sends out a LC64 response message to a MODCMS.
- After the time-out of an active LC64 value, the LC64 CMS goes to an LC64 CREATION/SELECTION state for reconfigurations.

and,

- An LC64.REQUEST is a transition when a LC64 capability request comes from a MODCMS.
- A NEW_LC64 is a transition when a new LC64 capability has been successfully created/selected.
- A PATH ACTIVE transition triggers a WAITING/DWELLING state.
- An LC64.EXPIRED transition happens when a current LC64 value is timed out.
- During the WAITING state, every LC64 request gets the currently active LC64 value through a LC64_RESPONSE transition.

Enterprise Access Routers

These routers provide upstream access to different topologically significant paths. Either a single router with multiple network interfaces can be connected to different upstream links or independent routers with access to individual upstream links can be used. Each topological path contains a different network namespace which is used as a LC64 capability.

²<http://ipset.netfilter.org/iptables.man.html>

4.2.3 LC64 Defence Methodology And Experiment Design

As LC64 defence is for the victim network to protect it from spoofing-based DoS attack, we can do an assessment about what happens to client communication during an attack while the LC64 defence is in place. The victim public servers represent the enterprise network so, we can do measurement on it rather than at the client side. It is expected that client traffic should be seen at the public victim servers with different destination L64 values. The idea is to move the victim network to a different upstream link than the one on which an attack is happening.

We form the following null hypothesis:

- *An LC64 defence does not increase client bandwidth, during a spoofing-based volumetric UDP flood attack, as compared to client bandwidth with defenceless victim site.*

In other words, there is no difference between true mean-bandwidth of client traffic in both cases.

Scenarios

The following are the three scenarios which are used for the empirical evaluation of LC64 defence:

- Scenario one — Client to victim communication without an attack and without a defence. This forms our baseline against which we compare the following two scenarios.
- Scenario two — Client to victim communication with a spoofing-based volumetric UDP flood attack in place.
- Scenario three — Client to victim communication with a spoofing-based volumetric UDP flood attack and a LC64 defence in place.

Testbed

The testbed consists of a DNS server machine with a collocated LC64 MOD-CMS, an LC64 CMS server machine, an enterprise server machine, a client machine, and an attacker machine. The testbed also consists of two vLANs, i.e., one data vLAN and one control vLAN. In the data vLAN we run the actual experiment while in the control vLAN we control the execution of the experiment and collect results.

The DNS server, client, attacker, LC64 CMS, and victim server have their own networks through respective routers shown in the Figure 4.5. Here, the operational assumption is that client and attacker are part of an external network that can access the enterprise network through its uplink (edge)

routers. The defence will work as long as there is a secret link between DNS and CMS (even if the DNS is in a different network). This setup can be generalized as long as all the client and attack traffic is coming from outside the enterprise network.

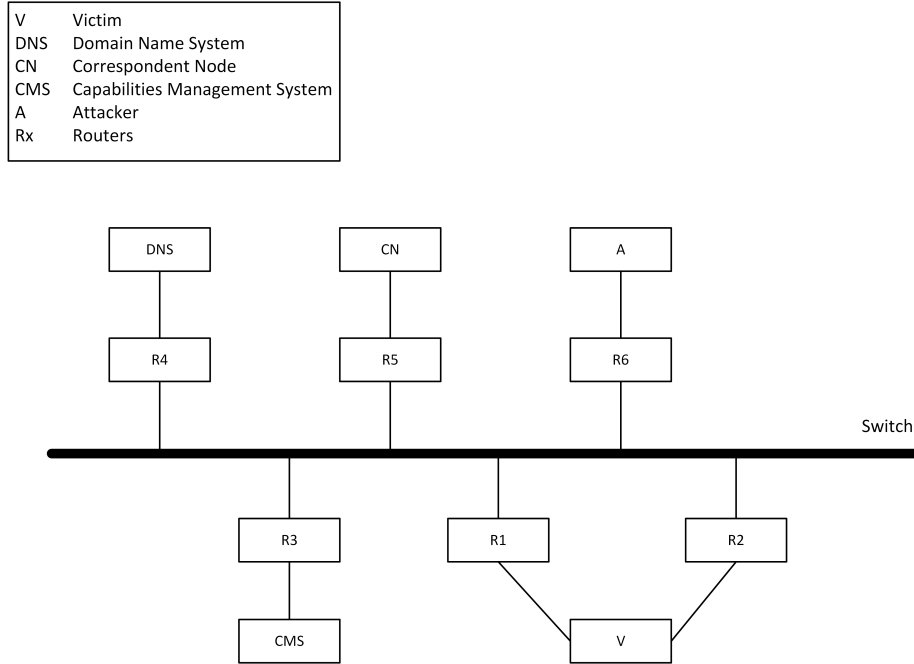


Figure 4.5: The testbed for an LC64 defence

Each router is directly connected to an Extreme[®] Switch x45a-48t with an isolated port, using 1 Gbps full-duplex Ethernet links. Each machine, apart from the switch in the testbed, is a Gateway[®] GR380 F1 machine with a 64-bit Intel[®] Xeon[®] 8-core CPU (2.27 GHz base frequency)³.

One of the interfaces in each router is part of the control vLAN. A different interface is used for the data vLAN. The victim is connected to two routers. These two routers act as upstream link providers with different network prefixes. The LC64 CMS controls routers R1 and R2. Each router is a Ubuntu 16.04 Linux box with packet forwarding turned on. The DNS server, and the LC64 CMS also run on an Ubuntu 16.04 Linux box. The victim and the client machines run an ILNP kernel.

We did not use any packet loss conditions because our defence depends on ILNP-specific Locator Update messages whose performance in the face of extreme network conditions has been extensively tested in [Phoomikiat-tisak, 2016]. This referenced research shows that Locator Updates have less overhead and packet loss than MIPv6 based solution.

³https://ark.intel.com/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-GHz-5_86-GTs-Intel-QPI

Experiment Configurations

Knotdns v2.4.1 acts as a forwarding DNS server with LC64 MODCMS bundled in as a module. The functionality of Knotdns to receive DNS requests and respond with DNS replies is not modified. MODCMS is only used as an interceptor of DNS requests. The victim and client machines run ILNPv6 kernel v4.9.38.

The routers R1 and R2 run an unmodified stable version 2.17 of radvd⁴. The maximum time allowed between sending router advertisements from radvd is set to 0.75 seconds in the radvd configuration. It should be noted that it is the maximum time, so, it is expected that as soon as the router sees change in its configuration it should send out an advertisement. ILNPv6 kernel sets the L64 namespace lifetime in the ILCC (see §2.1.2.1) cache to be equal to the prefix lifetime advertised in the router advertisement. So, as soon as one router stops sending Routing Advertisements (RAs), that particular L64 namespace expires in the ILNPv6 kernel.

The victim runs iperf3⁵ in server mode and the client runs iperf3 in client mode. The LC64 CMS machine runs a LC64-specific CMS daemon for capability management. The attacker machine uses iperf3 to generate a maximum possible bandwidth (~1 Gbps) of a continuous stream of UDP packets as a form of an attack.

The data is collected on the victim machine using the tcpstat⁶ tool. The tcpstat tool gives throughput measurements in bytes per second of client traffic (which we require) among other statistics.

Performance Metric For Evaluation

We performed the following measurement to assess the effectiveness of the LC64 defence. These measurements are independently taken for the three scenarios mentioned in §4.2.3.

- *Bytes per second:* When there is no attack, we will see a particular data rate in bytes per second of client traffic at the victim side. During an attack, if this data-rate is reduced then, it means that the attack has successfully reached a victim network through its router (the attacker's path is separate from the client path up until the access router of the victim). During an attack, while the LC64 defence is in place, if we see an increase in the data-rate of the client traffic compared to what it was when there was no solution, then we would be able to measure the effectiveness of the LC64 defence.

⁴<http://www.litech.org/radvd/>

⁵<https://iperf.fr>

⁶<http://manpages.ubuntu.com/manpages/trusty/man1/tcpstat.1.html>

Testing

We ran 25 iterations of each scenario based on the statistical power analysis to find the sample size (the number of individual non-overlapping measurements each with an independent experiment with same configuration) in order for the results to be significant. Each iteration was of 20 minutes duration. We chose the 20 minutes for two reasons. Firstly, it will allow us to diminish the affects of TCP slow start for client traffic. Secondly, most of the DoS attacks are under a few hours duration (Kaspersky Labs Q1 2018 report [[Kaspersky Labs](#),]). The 20 minutes duration per run also gives us, on average, a controlled data-rate of a UDP flood for each run. We used a 1 Gbps bandwidth for the iperf3 client using the UDP configuration. We also use a guard of 5 minutes in the beginning and the end of the experiment. This guard ensured that the start up and tear down phases of the experiment had no effect on the actual measurements.

Limitations And Validations

Our objective is to evaluate whether a LC64 defence can mitigate spoofing-based volumetric UDP DoS attack or not. It can be tempting that we test this solution under MAN and WAN emulated environments as we tested it in the case of NC64 defence but we show that it is enough to show an increase in the client data rate using a LAN environment only. [[Phoomikiattisak, 2016](#)] shows empirical evaluation of network mobility through the use of Locator Updates (LUs) under different network conditions. Instead of repeating the same experiment along with the same traffic profile (iperf3), we can show that a LC64 CMS is able to transition the whole site to a different path controlled by our backend. Once a network change occurs, any traffic will always follow the network conditions since the LC64 CMS will have no other role than to update the router interface from a separate path where there is no attack/client traffic. This is a significant observation which allows us to come to a conclusion using only a LAN environment.

The limitations and validations for the engineering of LC64 CMS and MODCMS daemons are the same as those for NC64 (please see §3.3.1).

4.2.4 Results And Statistical Analysis

The Figure 4.6 shows results of the scenarios mentioned in §4.2.3. Our objective is to regain client traffic once the LC64 defence is active.

In each iteration, we also took the mean of a client's throughput data as measured at the victim. Figure 4.6 shows throughput distribution in all iterations with a diamond symbol indicating the mean.

As the mean and median values were similar, it was beneficial to carry out further statistical analysis using a normal distribution. The 98th percentile was used to generate a qqnorm plot shown in Figure 4.7. The *qqnorm*

plot also gives us a plausible visual assessment of the nature of the distribution of our data. Linear trends in qqnorm show that the data of the results is normally distributed.

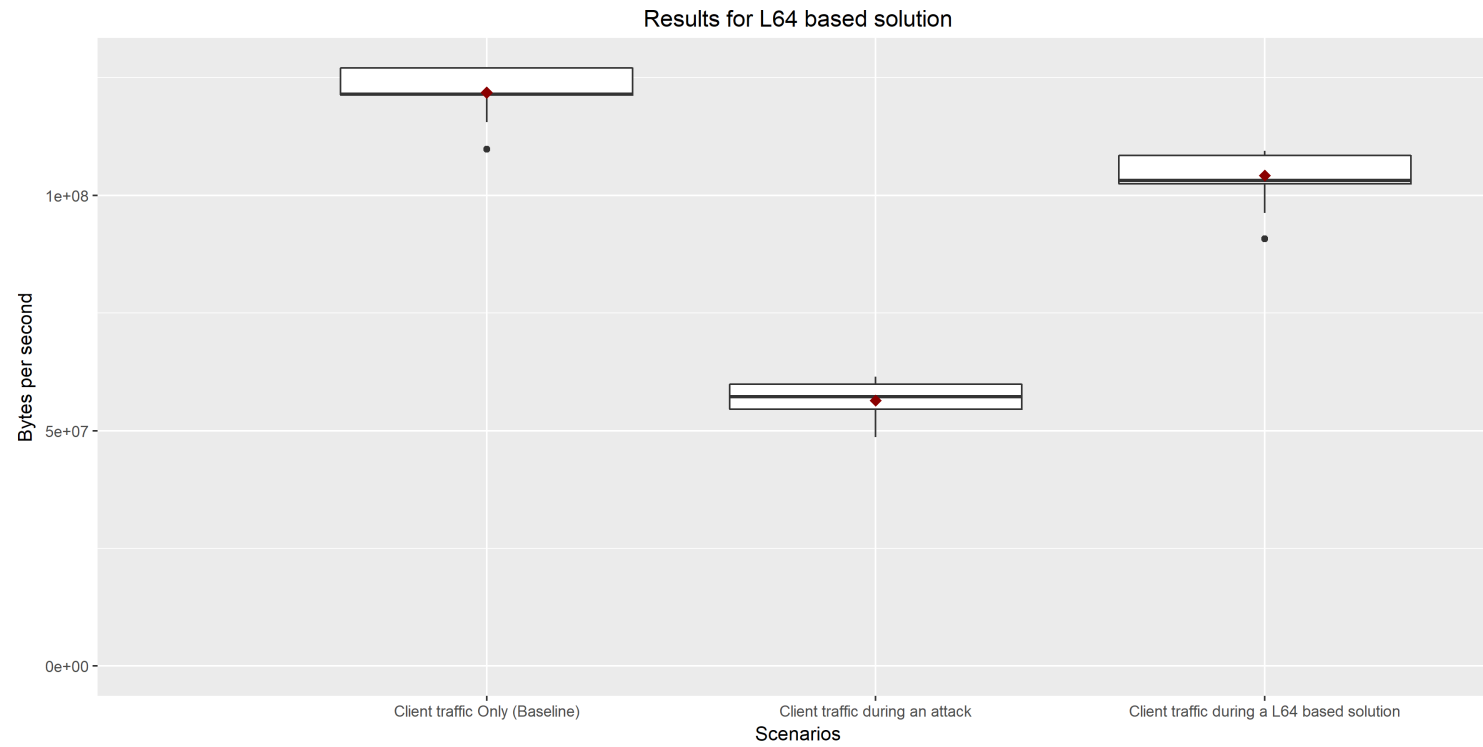


Figure 4.6: Traffic comparison under DDoS with an LC64 defence is in place. The diamond symbol shows the mean of all the iterations.

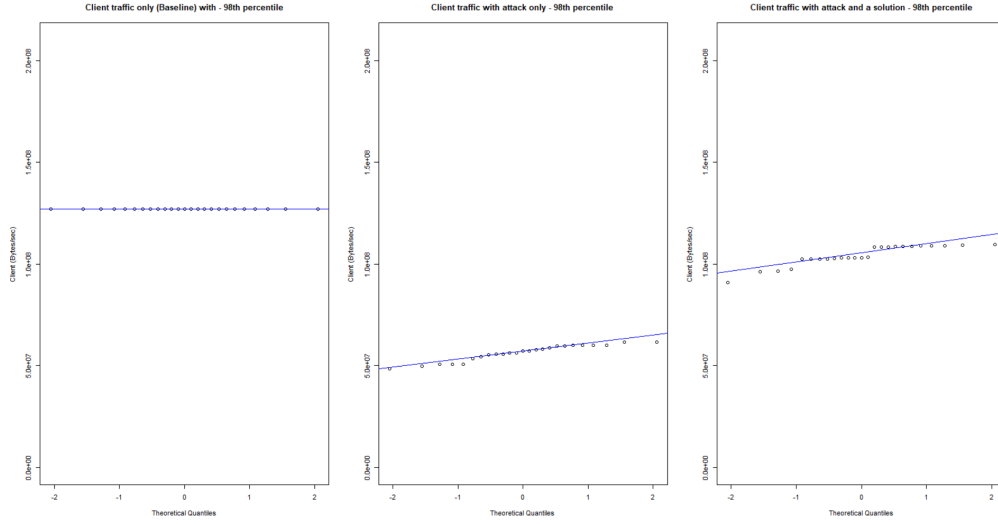


Figure 4.7: Quantile-quantile plots for all sub-scenarios in the L64 based defence

To calculate a p-value and to test the null hypothesis the *Welch two-sample t-test* was used, because we have to compare the true means of the client traffic during an attack but without a solution; and client traffic during an attack but with a solution in place. We applied the two sample t-tests on the effects of an attack on the client while comparing the baseline with an attack scenario. Similarly, we applied the two sample t-tests on the effects of the L64 defence on the attack scenarios with no defence.

Table 4.1 shows a p-value of $< 2.2e-16$ in the *LC64 Solution vs Attack* column, which allows us to reject the null hypothesis (see initial part of §4.2.3) and accept the alternate hypothesis.

Scenario/Statistics	Baseline vs Attack	LC64 Solution vs Attack
Mean (Client traffic)	~127.14 MB/s and ~56.42 MB/s	~104.22 MB/s and ~56.42 MB/s
p-value	$< 2.2e-16$	$< 2.2e-16$
Confidence Interval (99th Percentile) - Client Traffic	~68.56 MB/s to ~72.89 MB/s	~44.40 MB/s to ~51.20 MB/s

Table 4.1: This table shows the comparison of baseline with an attack-only scenario. It also shows comparison of the attack-only scenario with the LC64 defence scenario. Each measurement is the client bandwidth in Megabytes per second (MB/s) as measured at the victim host.

Given the above results, it is proven that a LC64 defence does, significantly, increase the client bandwidth during a spoofing-based UDP flood attack.

Locator Updates Performance And Overhead

Figure 4.8 shows the performance of L64 Locator Updates in terms of the average Round Trip Time (RTT) and the average number of locator updates per run.

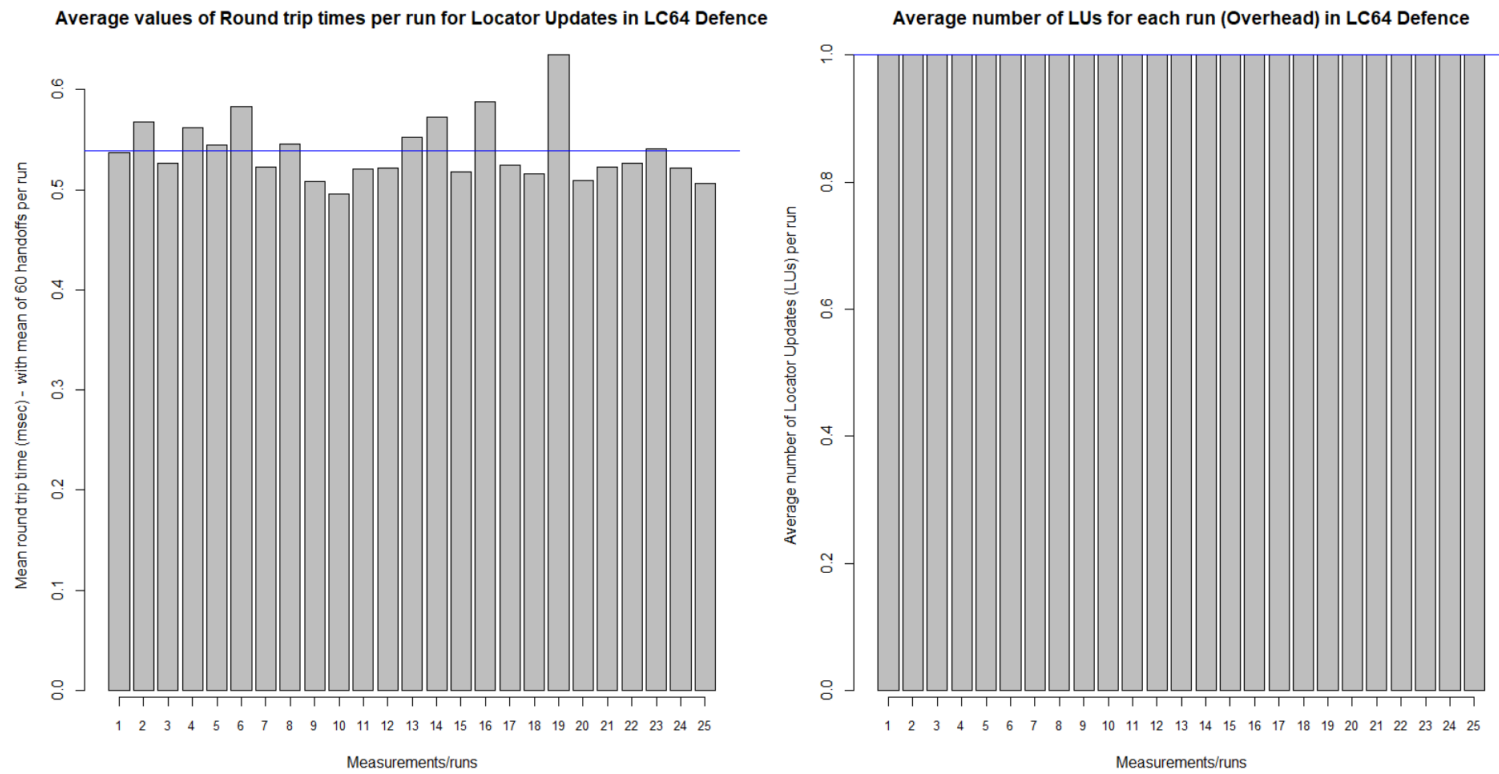


Figure 4.8: Locator Updates (LUs) performance and overhead

On the left side plot of Figure 4.8, we have the average RTT values per run/iteration of the experiment. There was a handoff every 20 seconds, so each iteration/run had 60 handoffs in each 20 minutes period. The expected RTT for a locator update includes a LU packet from the victim server and an LU-ACK packet from the client. So the output shown on the left of Figure 4.8 was an expected result.

On the right side plot of Figure 4.8, we have the average number of locator updates per run/iteration, which, effectively, gives us the overhead of locator updates. The average value of one implies that it took, on average, one LU message for the 60 handoffs in each 20 minutes period of each run. This average is expected and ideal (less LU retransmissions imply less overhead).

The blue horizontal lines, on the figure shows the mean values for all the 25 iterations.

4.3 Quantifying Attacker's Effort Displacement

As a LC64 defence uses DNS fast fluxing (see §1.6) to become a moving target for an attacker, it is important to measure reduction in the attack traffic when the solution is applied. The reduction in the attack traffic data rate is inversely proportional to an attacker's effort, i.e., an attacker has to generate more traffic on other downstream links, at the victim, to achieve a successful DoS attack. We term this as the attacker's effort displacement after starting the LC64 defence in the face of an attack.

To quantify the attacker's effort displacement, we designed another experiment with SYN-based volumetric DoS. We could have equally used UDP flood as well but for the sake of completeness, and to show that the solution works for high data-rate SYN flood attacks as well, we chose the SYN flood attack.

In this section our objectives are:

- Measure the change in attack traffic data-rate after the LC64 defence is enabled.
- Compare the change in attack traffic distribution before and after the LC64 defence is initiated/used.
- Analytically determine the effects of the *LC64 defence's scalability* on the attacker's effort in terms of the requirements to *increase the data-rate* to launch an effective DoS attack by the attacker.

4.3.1 Design Of The Experiment

The attacker's effort displacement is quantified in terms of the reduction of volumetric SYN flood traffic data-rate as seen at the victim network.

If the SYN flood traffic is reduced at the victim side, then the attacker's current attack traffic profile would be ineffective so as to force the attacker to increase the SYN flood traffic and attack on all the available locators (on all disparate topologically significant paths traversing through multiple regions of the Internet) to reach the same level of attack success when there was no defence.

The goal of the attacker is to cripple the availability of the victim's services to the client. We have covered this aspect in §4.2, i.e., using the LC64 defence to enhance availability of victim services to the client. Using the LC64 defence, we ensure that not only clients receive services from the victim but the attacker should be *discouraged* as well, in terms of the extra requirements, to increase the data-rate of DoS attack traffic. It is the second part which is the concern of this section's experiment.

The following two sub-experiments were done, each with 25 iterations (based on the statistical power analysis that gives us the number of iterations to run in order for the results to be significant).

- SYN flood traffic measurements without an active LC64 defence.
- SYN flood traffic measurements with an active LC64 defence.

The bandwidth for the duration of each iteration will be calculated which will give us 25 bandwidth measurements for each sub-experiment. Then, we statistically analyse and discuss the traffic distribution of each sub-experiment. We also, analytically, discuss the effects of LC64 scalability on the attacker's effort.

In our setup, the LC64 CMS used two L64 namespaces for DNS fast flux. Each L64 namespace is connected to two topologically separate upstream links. One L64 namespace, or ephemeral LC64 value, was active at one particular time; and the LC64 CMS changed to active locator after every 20 seconds using ILNP's host-based Locator Update (LU) path announcements.

4.3.2 Testing

The attacker generated volumetric SYN flood using the `thcsyn6`⁷ daemon, which generated ~0.4 million Packets Per Second (PPS). To test whether this affected the client traffic or not, we tested its effects on an `iperf3` flow from the client side. There was around 25 % reduction in client traffic. We then moved on to carry this experiment using ~0.4 million SYN PPS. Since we used a full-duplex 1 Gbps Ethernet link, the victim's access link was saturated with ~0.8 million PPS (~0.4 million SYN + ~0.4 million SYN-ACK (with extra overhead of retransmissions) packets).

We again emphasise that we did not use client traffic in this experiment because we wanted to see the traffic distribution of a DoS attack which was

⁷<https://salsa.debian.org/pkg-security-team/thc-ipv6>

100% effective. The ILNP packet along with TCP segment size was 80 Bytes in both directions, effectively giving us ~ 64 MB/s.

The Ethernet channel capacity of our testbed was ~ 125 MB/s in each direction. We did not employ another attacker machine to reach this capacity because we wanted to achieve attack traffic stability (the same data rate per iteration) in order to measure the effects of the LC64 defence on the attack traffic distribution. The scenario where Ethernet capacity was reached has already been tested when we measured the increase in client traffic during the LC64 defence in our previous experiment in this chapter. On the same note, Figure 4.8 showed the execution of ILNP's control traffic to counter DoS in the case where the channel capacity was exhausted.

tcpstat⁸ was used at the victim side to count SYN flood PPS before and after enabling the LC64 defence.

Figure 4.9 shows the testbed for this experiment.

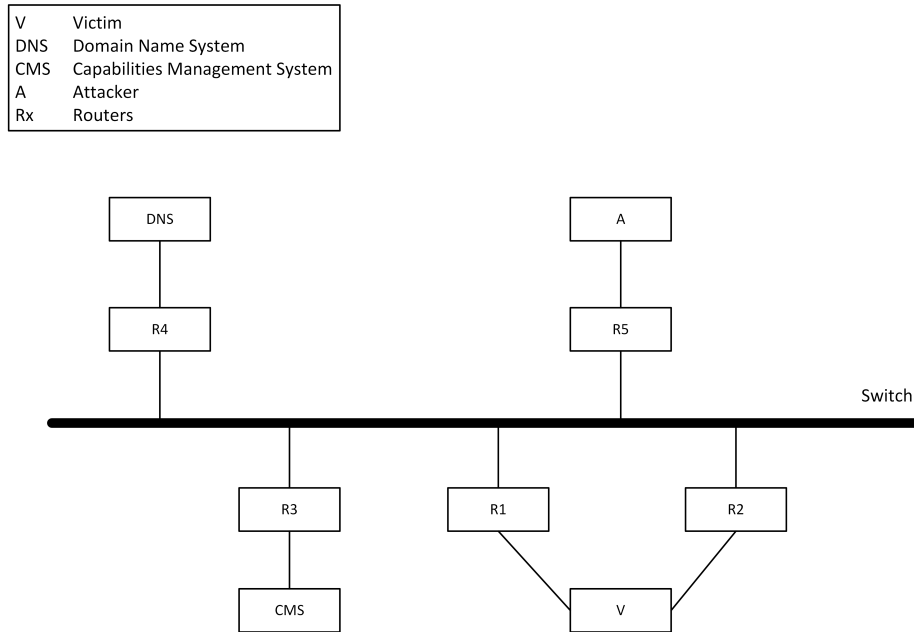


Figure 4.9: Testbed for LC64 defence to measure attacker's effort while doing fast flux on both the redundant link and the link under attack.

Each router is directly connected to an Extreme[®] Switch x45a-48t with an isolated port, using 1 Gbps full-duplex Ethernet links. Each machine, apart from the switch in the testbed is a Gateway[®] GR380 F1 machine with 64-bit Intel[®] Xeon[®] 8-core CPU (2.27 GHz base frequency)⁹.

⁸<https://www.frenchfries.net/paul/tcpstat/>

⁹https://ark.intel.com/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-GHz-5_86-GTs-Intel-QPI

Figure 4.10 shows the sequence diagram of the host/network interactions during fast fluxing among two locators including the path which is under attack. We included the path under attack in the fast flux to show attack traffic behaviour between locator transitions (we do not recommend inclusion of the attacked path in the DNS fast flux procedure in real world deployments).

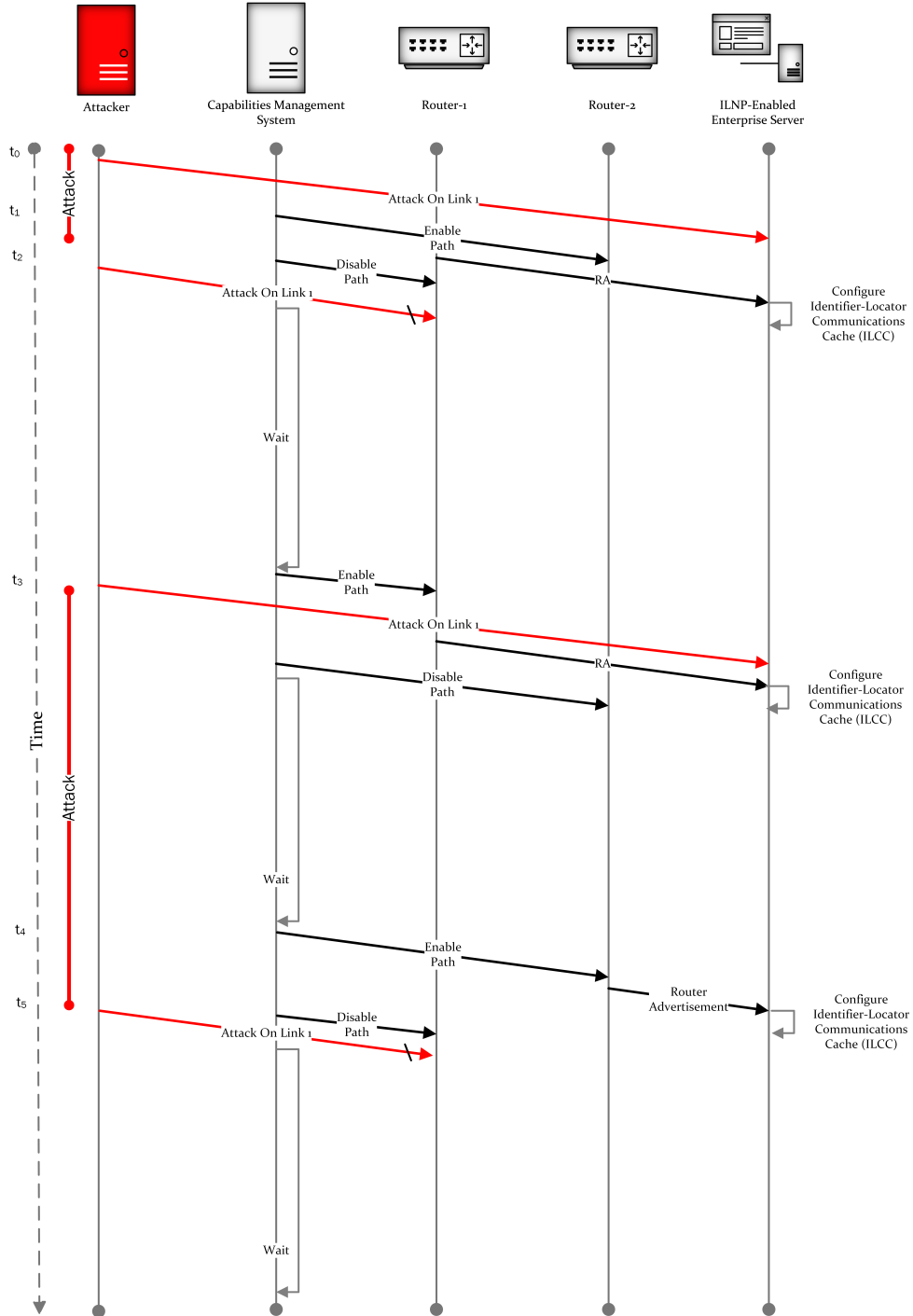


Figure 4.10: Sequence diagram for quantifying attacker's effort. It shows host/network interactions while the LC64 CMS fast fluxes between the link under attack (Router 1) and the redundant link (Router 2). RA is the Router Advertisement.

$t_x : x \in [0, \infty]$ is the time instance.

These interactions are:

- t_0 to t_2 : In this duration, an attack reaches the victim network because the upstream link on router 1 is still active. At t_1 , LC64 CMS enables the interface on router 2 which allows router 2 to send an RA to the victim host. Between t_1 and t_2 , the client and victim hosts would shift their communications to a redundant link using the LU mechanism (not shown since we are not using a client in this experiment). At t_2 , the LC64 CMS disables the link on router 1 so the attack traffic stops reaching the victim network.
- t_2 to t_3 : After disabling the link on router 1, the LC64 CMS waits for the duration of the respective active ephemeral LC64 capability. It waits until t_3 . During this period, there is no attack traffic seen on the victim host.
- t_3 to t_5 : At t_3 , LC64 CMS transitions the network to the old link and we start noticing the attack traffic at the victim host until t_5 at which point the LC64 CMS disables the link on router 1. Between t_4 and t_5 , the victim host can shift all the legitimate communications (if any) to an active path.
- t_5 onwards: We do not see any attack traffic at the victim side for the duration of the active ephemeral LC64 capability. After the waiting time (time duration of an active LC64 value), the LC64 CMS shifts to the other link. This switching takes place until the enterprise management decides to deactivate the defence.

4.3.3 Results

Figure 4.11 shows results for each scenario in this experiment.

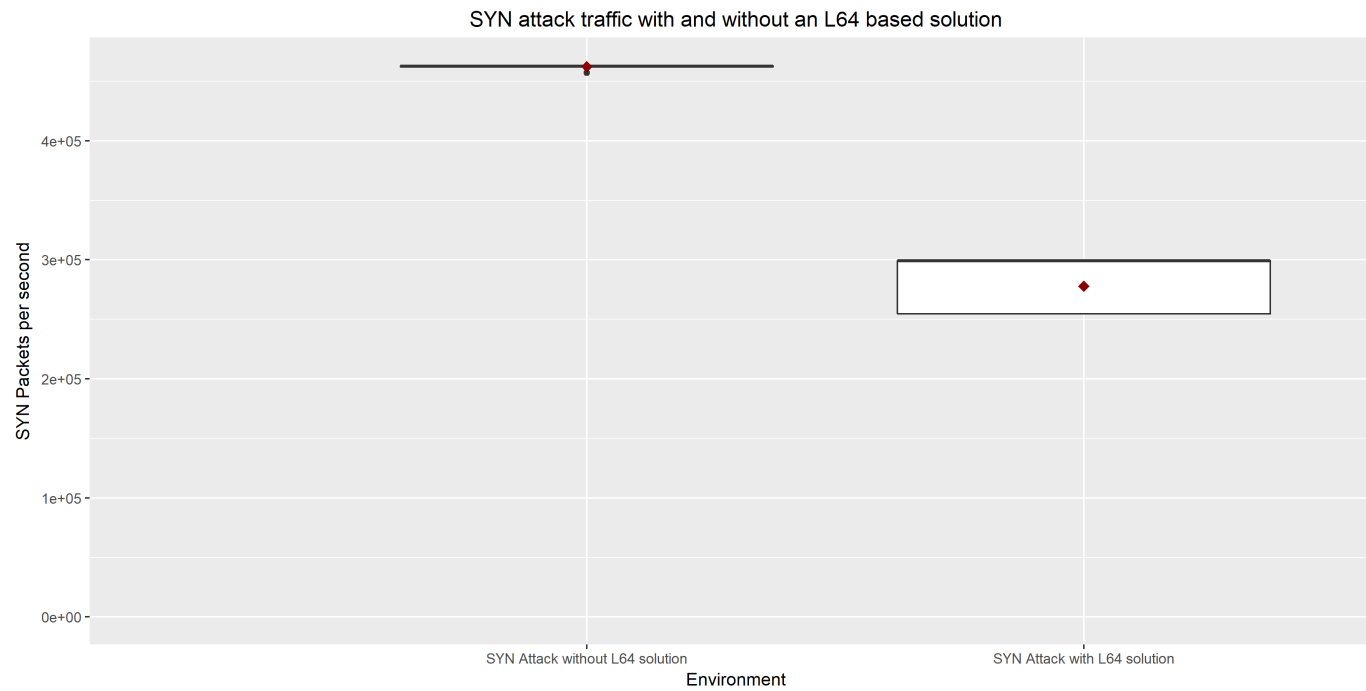


Figure 4.11: Boxplot showing displacement in attacker effort

SYN flood had a mean of ~ 0.46 million SYN flood packets per second before enabling the LC64 defence, and ~ 0.27 million packets per second after enabling the LC64 defence. This is $\sim 40\%$ reduction in SYN flood data rate. Some of the attack traffic reached the victim during the soft-handoff period (we set it to arbitrary 2 seconds to allow client connections to shift the network), so instead of seeing a 50% reduction we saw $\sim 40\%$ reduction. Given this observation, an attacker has to attack on the redundant link, as well, to achieve a successful attack on the whole site. This percentage can be reduced to zero if the LC64 CMS does not come back to or re-activate the link on which the attack is happening.

Equation (4.1) shows the change in attack data rate where R_b is the attack data-rate after enabling the defence, and R_a is the attack data-rate before enabling defence. β is a scaling factor greater than zero. In our testbed, with two ephemeral LC64 values, β tends to be ~ 0.40 . If the LC64 CMS shifts traffic to the redundant links and disables the attacked link for the duration of the attack, then β would be zero as the services would have been shifted to the redundant links and they will never come back to the attacked link unless the attack has disappeared.

$$R_b \simeq \beta R_a : \beta \geq 0 \quad (4.1)$$

4.4 Measuring SYN Flood Packets During L64 Transitions

A L64 namespace transition allows a change in the topological path, and it occurs when LC64 CMS sends interface/route configuration updates to upstream routers.

It is important to measure the volumetric attack traffic during a transition of an enterprise network to the redundant link, because it will give us information on how much attack traffic leakage occurs during this transition.

4.4.1 Design Of The Experiment

Figure 4.12 shows the topological diagram with a SYN attack coming from access router 1. This attack will run on the same subnet/prefix for the complete duration of the experiment. The duration of the experiment is 500 seconds with 20 seconds of guard at the beginning and the end. The SYN flood attack runs for the full duration of the experiment. After 120 seconds, we activate the LC64 defence. Each L64 transition occurs after every 20 seconds. We get a total of 17 transitions within the period from the 120th to the 460th second. We chose 17 transitions at random because it is already established that L64 transitions occur smoothly (see §4.2.4 and [Phoomikiattisak, 2016]).

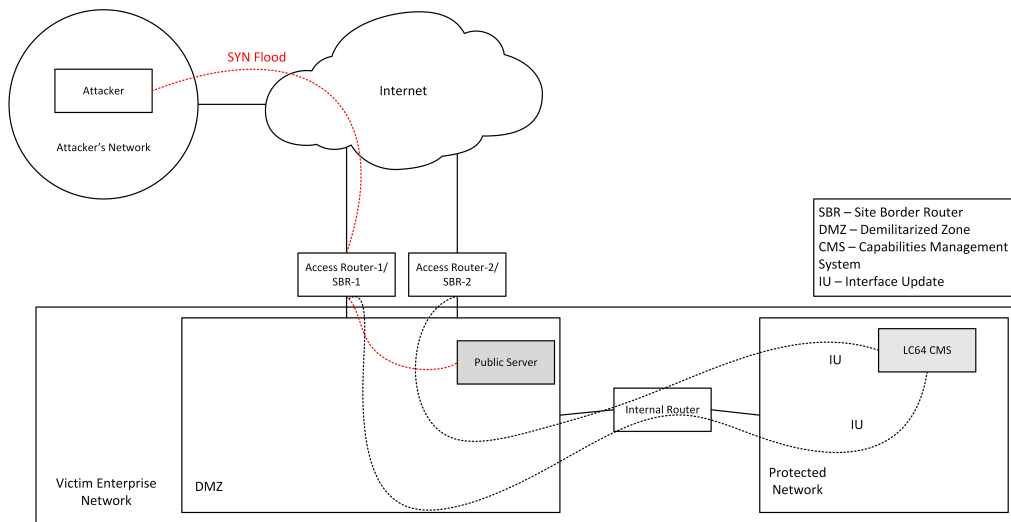


Figure 4.12: The logical diagram showing topological separation enabled by two uplink routers. SYN flood traffic comes through access router 1. The LC64 CMS controls the access routers using Interface Updates (IUs).

Figure 4.13 shows the experimental setup. Each router is directly connected to an Extreme[®] Switch x45a-48t with an isolated port using 1 Gbps full-duplex Ethernet links. Each machine, apart from the switch in the testbed, is a Gateway[®] GR380 F1 machine with 64-bit Intel[®] Xeon[®] 8-core CPU (2.27 GHz base frequency)¹⁰.

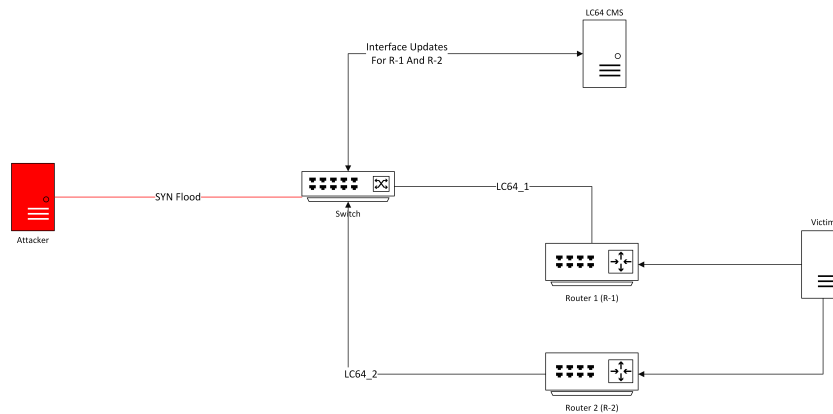


Figure 4.13: The testbed with topological separation

The victim machine runs the ILNP kernel version 4.9.38. The attacker, the LC64 CMS, and the routers runs on an Ubuntu server 16.04. The LC64

¹⁰https://ark.intel.com/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-GHz-5_86-GTs-Intel-QPI

CMS daemon is connected to the victim's access routers (two in this case) using SSH. Each network transition requires turning off the radvd daemon on the inactive access link, and invalidates routes using iptables. The second requirement during this switching is to enable radvd on an access router whose locator is about to get active, and to enable routing using iptables on that router. A script is used to perform the following steps in order:

1. Enable radvd and routing (using iptables) on the access router whose locator value will be active.
2. Disable radvd and routing (using iptables) on the access router(s) whose locator value(s) will be inactive.

We introduce a sleep of two seconds (just before disabling radvd and routing) at the access router whose locator is about to get in-active. This two second period can be reduced to the amount of time it takes for the desired locator to get active (based on an enterprise's own measurements and needs). Normally we would want to set this time to double the RTT of locator update, but for high packet loss environments it can be more than two seconds. We set it to an arbitrary 2 second period to show the transition and effects of incoming traffic while in soft handoff mode. A soft handoff is enabled on the victim. It is not a requirement but soft handoff actually enables us to have a large enough window for legitimate clients to receive a Locator Update as there might be packet retransmissions while the attack is still in place.

4.4.2 Testing

A SYN flood was generated with the same attack profile as described in section 4.3.2. tcpstat was used at the victim side to count SYN flood traffic in bytes per millisecond before and after enabling the solution. We chose millisecond granularity in order to see what happens to the attack traffic itself during a switch to another locator, i.e., while the LU mechanism is changing the topological path.

4.4.3 Results

Figure 4.14 shows information about what happens to attack traffic as seen at the victim. It is noted that the transition is abrupt just after a waiting time of two seconds. The green vertical lines show transitions to the first available locator (L64_1), and blue vertical lines show transitions to the other available locator (L64_2). The attacker uses L64_1 for attack traffic, and the victim is fast fluxing from one locator to another. We noticed that the transition is smooth (sub-second), and it allows legitimate traffic to pass through using locator updates (in our case, within two seconds).

It takes 4-10 milliseconds to enable `radvd` and activate routes on the active router. According to Figure 4.8, it takes, on average, 0.5 to 0.6 millisecond to do a successful locator update. We need to add these two time measurements and the network delays (e.g., LAN/MAN/WAN delays) to achieve a seamless transition. This shows that our two seconds interval after the transition can be reduced to 1 second or less. In this experiment, we only measured the delay of LC64 interface updates on routers and the amount of attack traffic during a transition. An instantaneous drop in attack traffic at the transition was expected as performing it only takes one LU and one LU-ACK to transition the network (see §4.2.4).

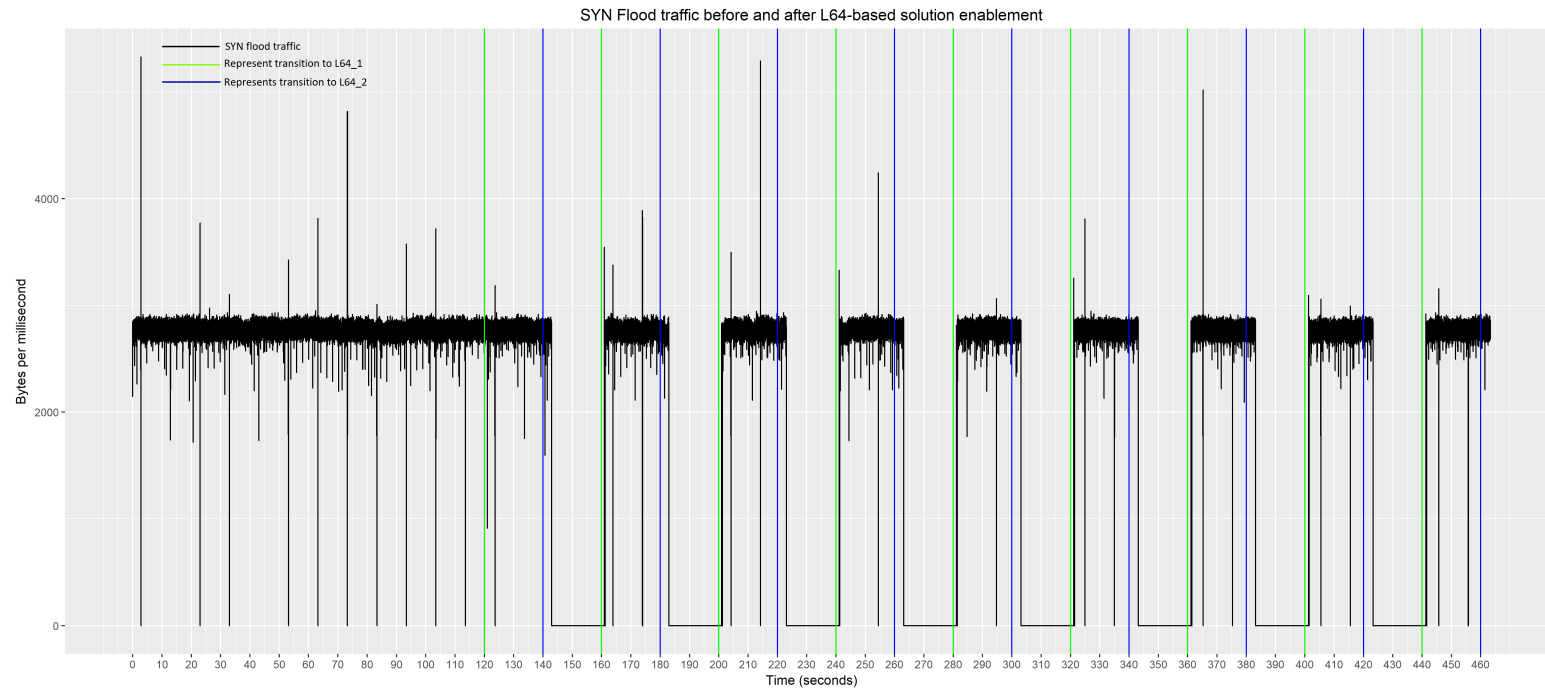


Figure 4.14: Time series showing fast flux between two locators and effects on SYN flood traffic

4.5 Summary

This chapter covered a novel contribution to defend an enterprise network against spoofing based volumetric DoS attacks (see §4.2). We used an ILNP locator namespace (see §2.1.1.2) to form capabilities which allow enterprise clients to shift their traffic from the attacked uplink to the safe uplinks. We termed these capabilities as L64 based capabilities (LC64) (see §4.1.1.1).

ILNP's first class feature of mobility (see §2.1.3), which is not supported by native IPv6, is employed along with its interaction with DNS. We also covered implementation details of the new infrastructure elements of the backend in our defence (see §4.2.2).

Empirical evaluation of the LC64 defence protocol was presented as a proof of concept. This chapter presented the design of the defence testbed, *baseline* against which the defence protocol was tested, assumptions made, limitations it had and their validations, its experiment configuration and execution, and statistical analysis of its results (see §4.2.4). We noticed that LC64 provides a viable solution to enterprise network security.

An important side-effect of the LC64 defence is that it allows an enterprise to restrict an attacker from vulnerability testing and probing of its infrastructure. This aspect will be discussed in chapter 6.

We presented an empirical evaluation of the displacement in attacker effort to launch a successful DoS attack (see §4.3). We noticed that after enabling the LC64 defence, the attacker's traffic was reduced to almost half as the client traffic spent half of the time on the safe uplink. This shows that an attacker has to attack all uplinks with more traffic than the amount of traffic it used when there was no solution in place.

This chapter also presented an empirical evaluation of the amount of attack traffic that leaks to the new uplink of the enterprise network when there is a transition at the end of the capability lifetime and the start of the lifetime of another capability (see §4.4). We noticed that the shift is almost instantaneous and there is almost no leakage.

Chapter 5

Multi-layered Enterprise Security And Client privacy Through ILNP DNS Capabilities

Chapters 3 and 4 evaluated DoS defences using the NID64 and L64 ILNP namespaces. We termed those defences as NC64 and LC64. NC64 defence protected the services running on an enterprise host while LC64 defence protected an enterprise network itself. In this chapter, we will provide an evaluation of a defence which is a combination of both approaches. We term this defence as LNC64 (L64 and NID64 Capabilities).

Normally, enterprises use more than one defence mechanisms to thwart DoS attacks. They might hire cloud scrubbing providers (see§2.3.7.2) or deploy security appliances at their edge network for traffic filtration (see§2.3.7.3). Multi vector attacks demand multi-layered security. We found that by combining our previous defences we can simultaneously provide security to an enterprise host and its site.

This chapter presents the following parts:

- Establishing a rationale for ILNP based LNC64 defence which can simultaneously protect an enterprise from low rate TCP SYN floods and volumetric UDP flood attacks. We should emphasise that the detection of each mechanism is neither a part of the rationale nor evaluation.
- Empirically evaluating the LNC64 defence with more than two redundant access links. As we have already evaluated NC64 and LC64 individually, we will only cover LNC64's feasibility during an attack along with its side effects. Our baseline will be defenceless client traffic with no attack in place.

- Presenting a novel concept of a prefix and subnet fast flux matrix which provides distinct sets of topologically separate and transient network paths for different enterprise hosts *outside* and/or *within* the same enterprise network.
- Evaluating possibility of client-privacy which is a side effect of applying ILNP-based defence along with the concept of a prefix and subnet fast flux matrix

5.1 Rationale

In Chapter 3, we compared the performance of NC64 and SYN cookies (see §3.3). We noted that NC64 performed similar to SYN cookies in MAN and WAN environments (see §§3.3.2.2 and 3.3.2.3). While NC64 is an effective solution for SYN flood defence, it is not effective for volumetric DoS attacks when the attacker is able to manipulate a victim's firewall.

LC64 protection against volumetric DoS attacks enabled the victim in our testbed to move from one topological location to another without disrupting the client session. We noted that LC64 defence allowed a client to improve its bandwidth (see §4.2.4). If an attacker can dynamically redirect its traffic to the victim's new network access links, then it can achieve DoS using a low-rate SYN flood as well. Otherwise, if it is using UDP flooding, then it has an option to increase its attack traffic on all the network access links (see §4.3).

If an enterprise wants to protect both its hosts and its network from both TCP SYN flood and UDP flood, then it requires a different approach. This chapter provides that approach.

L64 And NID64 Based Capabilities (LNC64)

A LNC64 defence provides ephemeral LNC64 values that contain a per-client NID64 value and a continuously changing L64 value for all clients within their respective sessions. If we have two clients and eight upstream links then, by using the NC64 mechanism (see §3.2.6), we can give NC64.1 to client one and NC64.2 to client two while making the victim transition from one LC64 value to another using the LC64 mechanism (see §4.2.1).

Figure 5.1 shows an ephemeral 128-bit LNC64 value formed using LC64 and NC64 values.

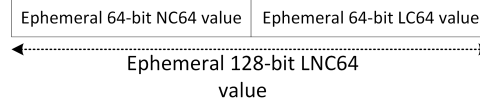


Figure 5.1: A 128-bit LNC64 value formed using a 64-bit LC64 value and a 64-bit NC64 value

Prefix Hopping Matrix With Overlapping Sequences

An L64 namespace and an IPv6 prefix have the same syntax and both of them can be used for routing using the longest-prefix matching [Trotter, 2001]. Given an LC64 defence can rotate among unique L64 namespaces, it is possible that we assign a single unique sequence of upstream links with separate paths to a single server within an enterprise. If we have eight upstream links then we can assign 40,320 permutations of eight L64 values to at most 40,320 servers within an enterprise. In such a case, the CMS would have to be modified to provide a different L64 value for a unique server through a DNS reply at one specific time and comply with a particular sequence allocated for that server. For example, at time t_1 , server one might have an L64_1 as its active locator while server two might have an L64_2 as an active locator. This effectively will give a specific sequence of L64 values to each enterprise server.

We can form a matrix of sequences where each cell contains a unique permutation of x number of L64 values. If an enterprise utilizes such a matrix, then we can make it difficult for an attacker to launch a successful DoS attack by limiting its efforts in network probing and vulnerability testing. Table 5.1 shows two servers with different L64 values being active at eight different times, i.e., utilizing two elements (sequences) from the matrix.

Time	t1	t2	t3	t4	t5	t6	t7	t8
Server One	L64_2	L64_1	L64_8	L64_4	L64_7	L64_5	L64_3	L64_6
Server Two	L64_8	L64_6	L64_1	L64_3	L64_4	L64_2	L64_7	L64_5

Table 5.1: Two servers are being allocated with two different sequences of L64 values. CMS allows them to hop from one locator to another. Each server might be using a different L64 value at one particular time.

Subnet Hopping Matrix

As L64 namespace and an IPv6 prefix have the same syntax and both of them can be used for routing using a longest-prefix matching [Trotter, 2001], an enterprise can use an ILNP subnetting mechanism [Atkinson & Bhatti, 2012b] to create different subnets among which individual victim hosts can perform DNS fast flux (see §1.6). It should be noted that, if an enterprise

has only one upstream link common to all subnets, then this approach does not protect an enterprise against volumetric DoS attacks. If the enterprise is using multiple topologically significant upstream links with disparate paths, then doing DNS fast flux on the complete range of L64 values will protect an enterprise from volumetric DoS attacks as well (see chapter 4). In any case, an enterprise can provide client privacy using this approach, which is a side effect of ILNP-based DNS fast flux mechanism (see §5.3).

Figure 5.2 (taken from RFC 6741) shows a locator subnet selector (L_{ss}) part which can be used to create multiple subnets within an enterprise.

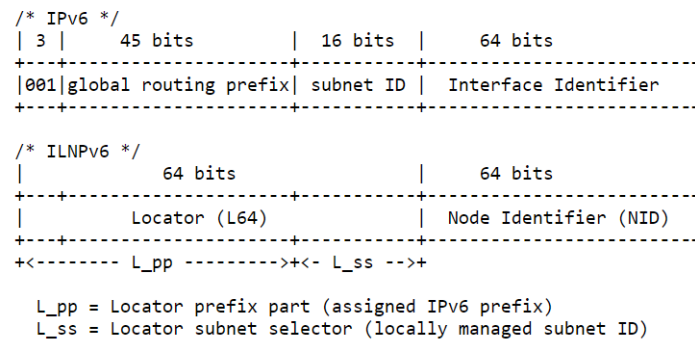


Figure 5.2: An ILNPv6 Subnet Identification in Comparison to an IPv6 Subnet Identification - Source: RFC 6741 [Atkinson & Bhatti, 2012b]

If an enterprise has been allocated a global locator prefix part (L_{pp}) or IPv6 prefix of /58, e.g.,

2001:0db8:3c4d:0015:00ba:cf00::/58

then an enterprise would have an option to use the remaining 6 bits (L_{ss}) to create $2^6 = 64$ locally-managed subnets. So, we make an 8 x 8 (“8 by 8”) matrix where we assign 8 different subnets to 8 different victim hosts within an enterprise. Figure 5.3 shows this matrix with an assignment of 16 locally-managed subnets to two hosts (eight subnets each). These subnets can be selected randomly or traffic engineering requirements of an enterprise.

00	01	02	03	04	05	06	07	Victim Host 1 Victim Host 2 Victim Host 3 Victim Host 4 Victim Host 5 Victim Host 6 Victim Host 7 Victim Host 8
08	09	0a	0b	0c	0d	0e	0f	
10	11	12	13	14	15	16	17	
18	19	1a	1b	1c	1d	1e	1f	
20	21	22	23	24	25	26	27	
28	29	2a	2b	2c	2d	2e	2f	
30	31	32	33	34	35	36	37	
38	39	3a	3b	3c	3d	3e	3f	

Figure 5.3: A matrix showing 64 different hex values formed using 6-bits of L_{ss}. Each hex value represents a subnet. Victim one uses subnets coloured as gold, and victim two uses the subnets shown in a dark grey colour.

To provide a defence against volumetric DoS attacks, it is a requirement for an enterprise to use one or more spare upstream links where the LC64 defence can shift all existing legitimate communications. If an enterprise wants to employ only one upstream link, then it has the option to provide client privacy through subnet hopping based on its business goals. In the prior case, it has both options (i.e., DoS defence and the possibility of client privacy).

Combining Prefix And Subnet Hopping Matrices

A prefix-hopping matrix allows an enterprise to have different but overlapping sequences from a matrix of at most $P(x, x)$ permutations, where x is the number of upstream links. Similarly, if there are y subnet bits in each upstream-provided L64 namespace/prefix (L_pp), then an enterprise can have 2^y subnets. If we combine all these, then an enterprise can assign a unique subnet to a server while performing rotations on upstream links based on a particular sequence from the prefix-hopping matrix.

The methodology presented here with prefix and subnet-hopping matrices can be used to introduce certain traffic engineering capabilities. Two of the possibilities are the DoS mitigation and the possibility of client privacy (see §5.3). It can be investigated further in future.

The following sections provide empirical evaluations of an LNC64 defence to enable enterprise security against DoS attacks. We will also provide an emulation to demonstrate that network layer client privacy is possible through the security mechanism provided in the LC64 and LNC64 defences.

5.2 Empirical Evaluation Of LNC64 DoS Defence

Figure 5.4 provides a logical diagram of an enterprise network with the required components for a LNC64 defence.

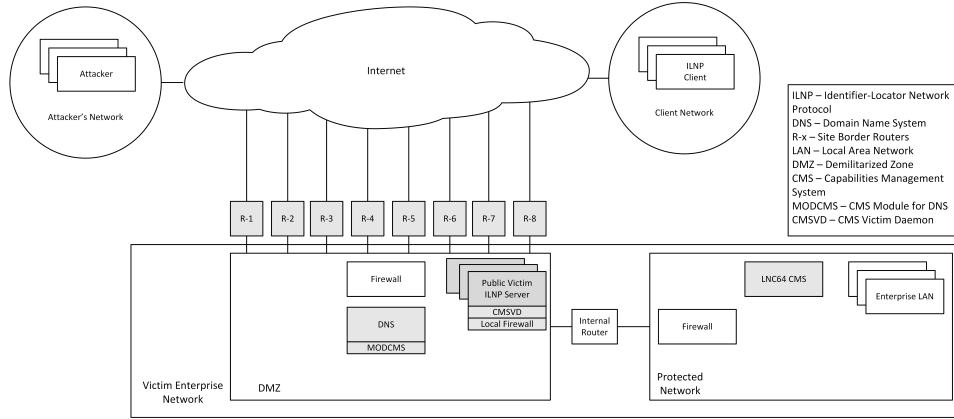


Figure 5.4: A logical diagram of an enterprise with eight upstream links to the Internet using eight routers, LNC64 CMS, CMSVD, MODCMS, DNS, public victim servers, client network, and victim network.

An enterprise is connected to the Internet through eight upstream links using eight routers but each link must be topologically significant and must offer a path that is not part of the path being attacked by DoS. We note that in real networks, enterprises have multiple links to the Internet so we utilize this feature. The following components are used within the network to enable a LNC64 defence.

DNS:

DNS is responsible for forwarding DNS name resolution requests from clients to MODCMS (described below). In our evaluation, we used KnotDNS¹ as the DNS server software.

MODCMS:

MODCMS is a LNC64 DNS module which communicates with DNS and the LNC64 CMS (described below). It takes a forwarded query from DNS, extracts client information, e.g., source address, creates and sends a LNC64 capability request message to LNC64 CMS, receives LNC64 capability responses from the LNC64 CMS, and forwards a DNS response to the DNS which, in turn, sends a DNS name resolution response to the client.

A comprehensive description along with finite state machine of MODCMS can be found in §§3.2.1.2 and 3.2.5.2.

¹<https://www.knot-dns.cz>

LNC64 CMS And LNC64 Defence Modes:

The LNC64 CMS combines functionalities provided by the NC64 CMS and the LC64 CMS. The LNC64 CMS communicates with the MODCMS, victim host, and upstream routers, and it simultaneously operates in the following two modes:

- *NC64 mode:* Upon reception of a LNC64 capability request from MODCMS, it creates an ephemeral NID64 value, puts it inside a NC64 MAP (see §3.2.3 and appendix A for MAP description), and sends this mapping to the victim host where the victim host can configure its firewall. The victim sends a map acknowledgement message to the LNC64 CMS which in turn sends a capability response to MODCMS containing the ephemeral NID64 value.
- *LC64 mode:* Once the LNC64 CMS receives a map acknowledgement from the victim host, it enables the LC64 mode. In the LC64 mode, it communicates with the upstream routers by enabling/disabling routes based on a schedule mandated by the enterprise management. In this mode, it follows all the dynamics of a LC64 defence (see chapter 4).

We combine the FSMs of the NC64 and LC64 defences without changing any previously defined custom messages among different parts of the system. The interaction between these two modes is enabled whenever a MAP acknowledgement is received from the victim. Upon this message a shift to the LC64 defence occurs.

Figure 5.5 shows a simplified finite state machine of LNC64 CMS with function names and respective input/output signalling flows.

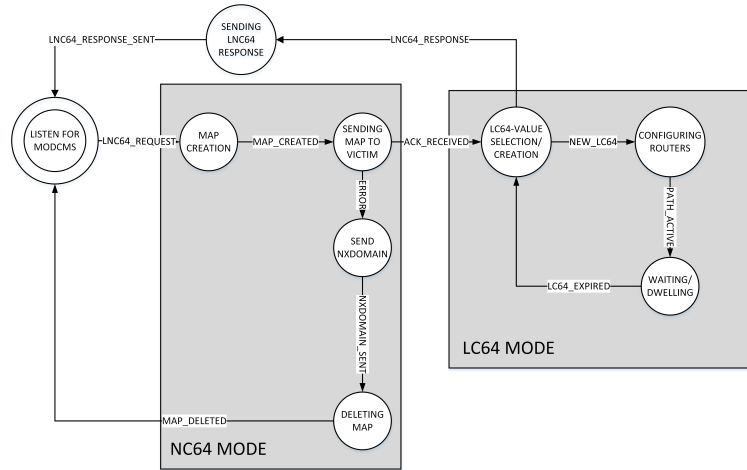


Figure 5.5: The finite state machine of an LNC64 CMS.

Where,

- LISTEN FOR MODCMS is a state where the CMS is listening for LNC64 capability requests from MODCMS.
- MAP CREATION is a state where the CMS creates a new NC64 capability, and maps it to the client information.
- SENDING MAP TO VICTIM is a state which is executed after a new NC64 capability is created. In this state, it sends the new MAP to the CMSVD at the victim host.
- If LNC64 CMS receives an ERROR condition from CMSVD, it goes to the SEND NXDOMAIN state where it sends the NXDOMAIN message to MODCMS which in turn delegates the NXDOMAIN response to DNS. Then, the DNS can send a NXDOMAIN DNS response back to the client. Once the NXDOMAIN message has been sent to the MODCMS, the LNC64 CMS goes to the DELETING MAP state.
- In the DELETING MAP state, the CMS deletes its local MAP. Once it has deleted the local MAP, it then goes to the LISTEN FOR MODCMS state again.
- If LNC64 CMS receives a MAP acknowledgement from CMSVD, it goes into the LC64-VALUE SELECTION/CREATION state where it selects/creates an active LC64 value.
- In the LC64-VALUE SELECTION/CREATION state, the CMS produces two messages. If it selects an active LNC64 capability, then that value is sent to the client through the SENDING LNC64 RESPONSE state. If it creates a new LC64 capability, then it will go into the CONFIGURING ROUTERS state where it enables the active path on one router and disables the previously active path on the second router.
- Once the routers are configured, it goes into WAITING/DWELLING state where it starts a timer for deactivation of the current active LC64 capability.
- Once the previous timer expires, it goes again into the LC64-VALUE SELECTION/CREATION state.

and,

- LNC64_REQUEST is a transition when the LNC64 capability request comes from MODCMS.
- MAP_CREATED is a transition when the LNC64 CMS successfully creates an NC64 MAP.

- `ACK_RECEIVED` is a transition where the LNC64 CMS receives an acknowledgement from CMSVD that a NC64 MAP has been successfully installed in the firewall of the victim host.
- `ERROR` is a transition where the LNC64 CMS receives an error message from the CMSVD that there was a problem in the firewall rule installation.
- `NXDOMAIN_SENT` is a transition where the LNC64 CMS successfully sends an `NXDOMAIN` message to the MODCMS.
- `LNC64_RESPONSE` is a transition where a LNC64 capability response message has been successfully sent to MODCMS.
- `NEW_LC64` is a transition where it selects a newly active LC64 capability.
- `PATH_ACTIVE` is a transition where all the routers have been reconfigured.
- `LC64_EXPIRED` is a transition where the timer associated with an active LC64 capability ends.

CMSVD

CMSVD is a victim host daemon which communicates with the LNC64 CMS. It is responsible for receiving NC64 MAP requests, installing firewall rules based on these requests, and sending acknowledgement/error messages to the LNC64 CMS. It is the same software that was used in the NC64 defence (see chapter 3). A comprehensive description along with a finite state machine of CMSVD can be found in §§3.2.1.3 and 3.2.5.3.

5.2.1 Experiment Design

We use a single access router whose nine interfaces are connected to nine interfaces on the switch. It is also connected to a victim host using a single link. Eight interfaces are used by the defence, and the ninth is used for a client traffic only scenario (see §5.2.1). It is assumed that this ninth interface will be attacked and the LNC64 defence will perform fast fluxing among the other eight interfaces. Each interface emulates an upstream link to the Internet, and is configured with a unique L64 value. Each link is an isolated link and does not affect any of the traffic on other links.

The LNC64 CMS is directly connected to the DNS/MODCMS, the access router, and the victim host. The LNC64 CMS connection to the router is on a different interface not accessible through the victim host.

The victim host and client machines are ILNP-enabled as it is a requirement for them to be ILNP-capable because of ILNP namespaces being in

use (see §5.2). All other machines are running an IPv6 kernel because there is no requirement for them to be upgraded.

Ethernet links among the machines support 1 Gbps full duplex mode. We used netem² to emulate various delay environments with varying packet loss conditions. We emulated three environments with delay of 0 ms, 25 ms and 210 ms. We used 2.5% and 5% packet loss in each direction. For example, the link between the client and router will be enabled with 5% or 10% end-to-end packet loss based on an individual network environment and scenario. We chose such delay and packet loss conditions because they are expected to occur while a volumetric DoS attack is in progress and this attack might fluctuate the client traffic between similar delays and packet losses.

Performance Metric For Measurements

We will use bandwidth measurements on each locator to see if traffic is distributed among multiple locators. A client is given a specific capability, and then the solution uses eight locators to fast flux among them. A traffic monitor is used at the victim host to measure the bandwidth used within each locator.

Scenarios

The following are the two scenarios which are used for the empirical evaluation of the LNC64 defence:

- Scenario one — Client to victim communication without LNC64 defence. This forms our baseline against which we compare with the following scenario.
- Scenario two — Client to victim communication with a spoofing-based volumetric UDP flood attack and a LNC64 mechanism in place.

Note that we did not form the third scenario of an attack while the client to server communication is happening with no solution in place, because it is already established that the client does see service disruptions/degradations while a volumetric attack is in place (see §§3.3.2 and 4.2.4).

Scenario two uses UDP flooding, whereas we could have equally employed TCP SYN flooding. We assume that a SYN flood will be mitigated due to inclusion of the NC64 mode within an LNC64 defence. If we had only used low or high data-rate SYN flooding, then a similar assumption could have been made for UDP flooding. We checked the victim host firewall rules during the execution of the experiment to ensure that the NC64 defence will honour per-client capabilities.

²<https://wiki.linuxfoundation.org/networking/netem>

Experiment Configuration

The DNS server is installed with KnotDNS v2.4.1, and the MODCMS is collocated within the DNS server as a KnotDNS module. The client and the victim hosts run with an ILNP kernel v4.9.38 with Ubuntu Server 16.04 distribution. The LNC64 CMS, the DNS server, and the attacker machines use an unmodified Linux kernel with an Ubuntu 16.04 distribution. Each router is a Ubuntu 16.04 server with packet forwarding enabled, radvd v2.17 installed, and a custom script (written in bash³ v4.3.48) which takes commands from the LNC64 CMS through a SSH-enabled link to update radvd configurations and upstream paths. The maximum time allowed between sending router advertisements from radvd is set to 0.75 seconds as is done by the LC64 defence. Any value less than one second is acceptable because ILCC (see §2.1.2.1) cache entries update active locator bindings after one second.

The CMSVD daemon is installed on the victim machine. The victim machine is using a nftables⁴ v0.8.3 kernel module as a firewall. The CMSVD installs NC64 MAPs in the nftables during the LNC64 defence execution.

The attacker machine runs iperf3 v3.1.3 with a bandwidth command line option set to 1024 Mbps so it can congest the Ethernet link to the victim. The client machine runs iperf3 v3.1.3 in client mode with a bandwidth command line option set to 1024 Mbps as well. The victim machine runs the same iperf3 software in server mode. Each Ethernet link supports 1 Gbps full-duplex mode.

Data is collected using tcpdump⁵ on the victim machine. The tcpdump tool will provide us with the bandwidth measurements required for performance evaluation.

Testing

Figure 5.6 shows the testbed used to evaluate LNC64 defence.

³https://www.gnu.org/software/bash/manual/html_node/index.html

⁴<https://netfilter.org/projects/nftables/>

⁵<https://www.tcpdump.org>

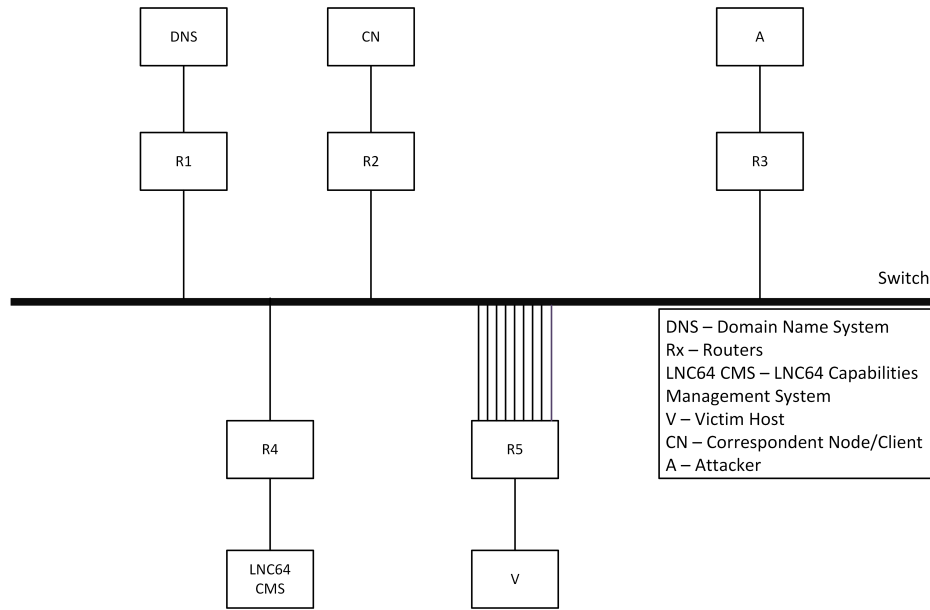


Figure 5.6: The testbed for evaluating the LNC64 defence. Nine interfaces of router five are connected to nine ports of the switch, providing unique and isolated upstream links. Each interface is configured with a unique L64 value. Router 5 acts as an upstream access router for the victim. The attacker attacks on the interface where the client communication takes place, before enabling the defence. Once the LNC64 defence is enabled, the rest of the eight locators are used by the defence.

The rest of the routers are also directly connected to an Extreme[®] Switch x45a-48t with isolated ports, using 1 Gbps Ethernet links. Each machine, apart from the switch in the testbed, is a Gateway[®] GR380 F1 machine with 64-bit Intel[®] Xeon[®] 8-core CPU (2.27 GHz base frequency)⁶.

We ran 25 iterations of each scenario based on statistical power analysis. Each iteration in each scenario was run for 160 seconds with 20 seconds of guard interval at both ends. The 20 seconds of guard interval ensures that the measured traffic will be stable and there will be no effects of TCP slow start on the final results. We ran each iteration of LNC64 for 160 seconds because with this duration we would be able to see a transition among every locator at-least once (the transition occurs after every 10 seconds).

It should be noted that scenario one and two were run with 20 minutes per iteration under the LC64 defence (see chapter 5). This does not affect our scenarios in this section, because in a LNC64 defence, we are concerned about the ability of a single host to shift communications among a set of distinct locators from the fast flux matrix.

⁶https://ark.intel.com/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-GHz-5_86-GTs-Intel-QPI

We also assume that after activating the LNC64 defence, the access router (i.e., R5) does not use the attacked link. In each iteration, a different NC64 capability was returned by the LNC64 CMS, although there is no requirement for it because the CMSVD ensures that the NC64 map is successfully installed and active in the victim host, for a single client, for the complete duration of the experiment.

5.2.2 Results

Figures 5.7 to 5.9 show the results for this section's evaluation.

The total traffic from scenario one is divided among a set of eight locators in scenario two.

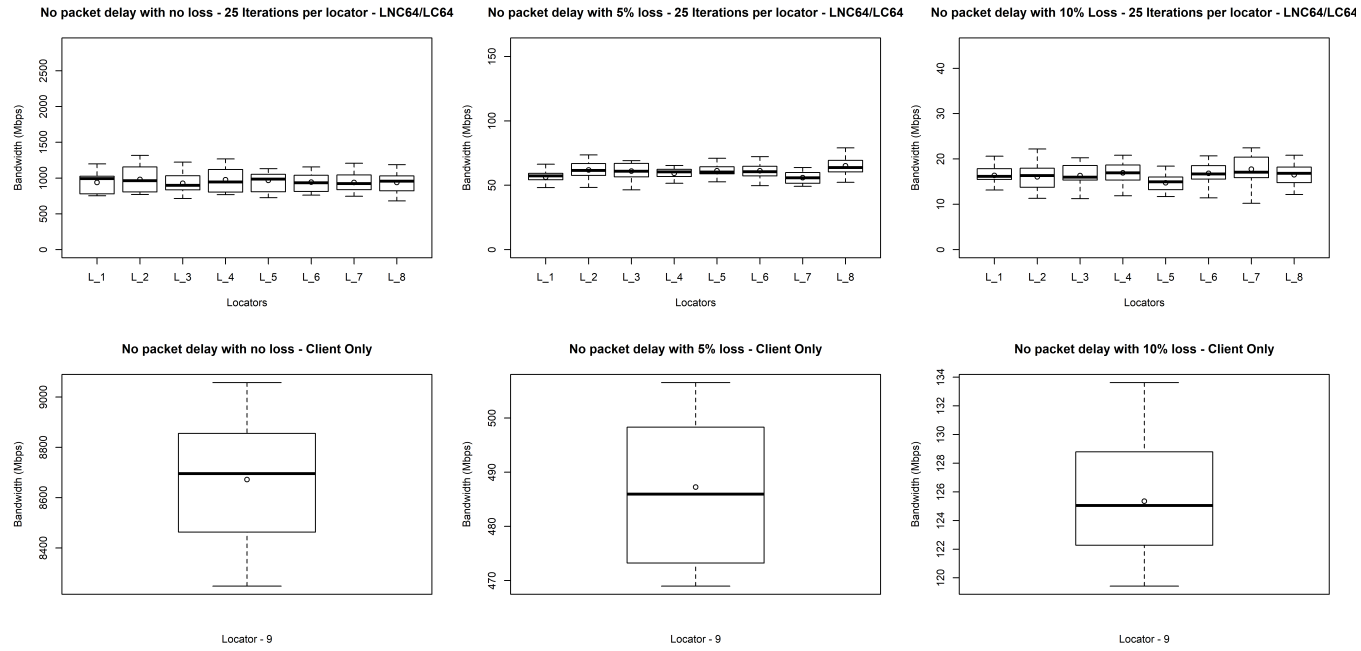


Figure 5.7: The results comparing the two scenarios with no packet delay. One session is run under the LNC64 defence (top row), and the second session is run without the LNC64 mechanism. It also shows the 5% and 10% emulated packet loss environments.

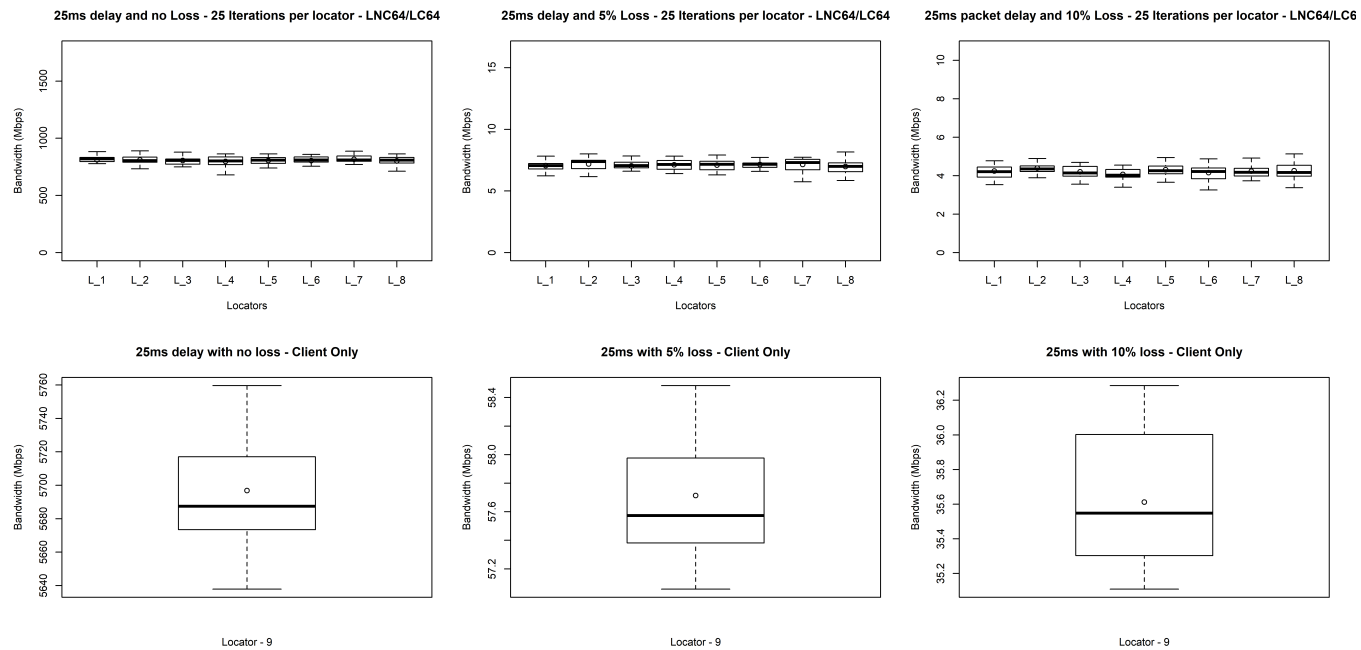


Figure 5.8: The results comparing the two scenarios with 25 ms end-to-end delay. One session is run under the LNC64 defence (top row), and the second session is run without the LNC64 mechanism. It also shows the 5% and 10% emulated packet loss environments.

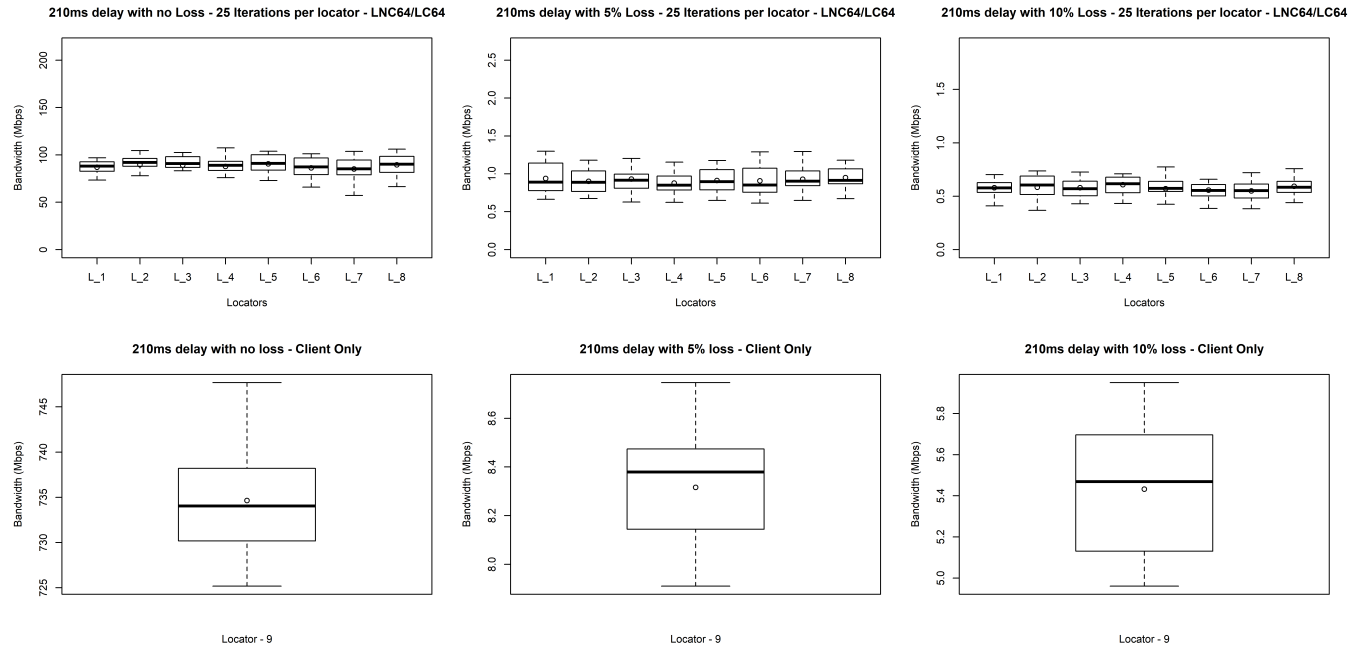


Figure 5.9: The results comparing the two scenarios with 210 ms end-to-end packet delays. One session is run under the LNC64 defence (top row), and the second session is run without the LNC64 defence. It also shows the 5% and 10% emulated packet loss environments.

Table 5.2 shows iperf3 communication for each scenario. Locator one through locator eight are used in scenario two while locator nine is used in scenario one. It can be seen that traffic coming on L9 is divided amongst all the locators in scenario two. The aggregate traffic in scenario two is missing some packets as we did not replay the same traffic in two scenarios but we ran iperf3 with the same configuration in both scenarios. It is not the missing packets we want to measure but we are concerned about iperf3 communication divided among a set of locators in different environments with the same/common software configurations.

Environment/Locators	L1	L2	L3	L4	L5	L6	L7	L8	L9
No delay - no loss	937	982	927	978	968	942	939	938	8672
No delay - 5% loss	56	62	61	59	61	61	56	65	487
No delay - 10% loss	16	16	16	17	15	17	18	17	125
25 ms delay - no loss	811	810	805	797	805	804	816	804	5696
25 ms delay - 5% loss	7	7	7	7	7	7	7	7	58
25 ms delay - 10% loss	4	4	4	4	4	4	4	4	36
210 ms delay - no loss	87	90	89	88	91	86	85	89	735
210 ms delay - 5% loss	1	1	1	1	1	1	1	1	8
210 ms delay - 10% loss	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	5

Table 5.2: The traffic comparison for the two scenarios with an average bandwidth of 25 runs in Mbps, where locator nine (shown in yellow) is only used for the client communications in scenario one, whereas the client will shift to other links in scenario two. We have rounded up the numbers to at most 4 significant digits.

In summary, the LNC64 defence enables an enterprise to effectively distribute the traffic among a set of upstream links without losing the end-to-end connectivity.

5.3 Client Privacy Side Effect Through LNC64/LC64 Mechanism

When a LNC64 defence is in use, an individual client session will see multiple ephemeral locators. Otherwise, a single fixed locator value will be seen for the session (if there is no other defence similar to LNC64 is active). If it becomes difficult for an eavesdropper to distinguish a particular session among multiple client sessions, some level of client privacy can be achieved as a side effect of the LNC64 DoS defence. This side-effect is possible if ILNP's control messages for mobility (see §2.1.3) are encrypted.

To show this side-effect, we made use of the ILNP subnet hopping matrix (see §5.1) with a Python-based emulation. We will again emphasise that,

if an enterprise is only using a subnet hopping matrix, then it will not be protected from DoS but it can provide some level of client privacy to its clients. We can also use prefix hopping matrix, which also protects against DoS attacks, but for emulation purposes any one of these can be used.

5.3.1 An Alternative To MPTCP ADD_ADDR Based Mechanism

Client privacy can also be achieved through the *path management* capabilities of the Multipath TCP (MPTCP) [Ford et al., 2013], e.g., using the ADD_ADDR TCP option which is used to advertise the creation of new sub-flows. Multipath TCP is an extension of the TCP protocol that allows a flow to be split into multiple sub-flows that can take different paths from within the network while maintaining the end-to-end goals of a communication, e.g., connection continuity.

It should be noted that ADD_ADDR can be used by an off-path attacker to put its own IP address in the ADD_ADDR TCP option, hence achieving connection hijacking. There are other residual threats associated with MPTCP [Bagnulo et al., 2015], some of which can be eliminated through the use of ILNP. Future studies can be devised in order to compare MPTCP (after fixing its threats) and ILNP with its protected LU.

As MPTCP uses TCP options for its path management capabilities, so it does not perform well if SYN Cookies are enabled because SYN Cookies are incompatible with TCP options (see §3.1). The MPTCP issues with SYN Cookies are also documented in [Bagnulo et al., 2015].

An MPTCP-based CMS can be designed which can provide the allocation and distribution of IP addresses formed using values in the sequences of prefix or subnet hopping matrix and host identifiers. Afterwards, that mechanism can be compared to the ILNP-based mechanism where LUs are encrypted.

The mechanism provided here through the LC64/LNC64 mechanisms can be used as an alternative to the ADD_ADDR-based MPTCP approach, only when the LUs are encrypted. This demands a future investigation and empirical comparison between the two.

5.3.2 Emulation Design

In §5.2, we made use of eight upstream links to the Internet, but in this section we assume that an enterprise has been given a single /58 global prefix (L_{pp}). An enterprise can use the remaining six bits of the 64-bit locator to create a locally-managed subnet hopping matrix as shown in Figure 5.3.

In this emulation, we will use the following distinct set of subnet values for one server:

$$S = \{38, 19, 2b, 3c, 0d, 16, 36, 3f\}$$

where each element in set S is a hex value of a single L_{ss} (see Figure 5.2).

00	01	02	03	04	05	06	07	Victim Host 1
08	09	0a	0b	0c	0d	0e	0f	Victim Host 2
10	11	12	13	14	15	16	17	Victim Host 3
18	19	1a	1b	1c	1d	1e	1f	Victim Host 4
20	21	22	23	24	25	26	27	Victim Host 5
28	29	2a	2b	2c	2d	2e	2f	Victim Host 6
30	31	32	33	34	35	36	37	Victim Host 7
38	39	3a	3b	3c	3d	3e	3f	Victim Host 8

Figure 5.10: A subnet hopping matrix showing the subnet slots used by the enterprise host in our emulation. Each subnet is shown as a hex value of six bits (L_{ss}).

Figure 5.11 shows an enterprise network with one /58 global prefix (L_{pp}) assigned to it. There is only one global topologically significant path, so, the LNC64 or LC64 mechanisms will not be able to protect the enterprise network from a volumetric DoS attack. We assume that there is no attack, and that ILNP control signalling is encrypted. If an enterprise has two /58 topologically significant global prefixes then it can use two subnet hopping matrices, so the availability of these matrices scales with the number of distinct upstream links.

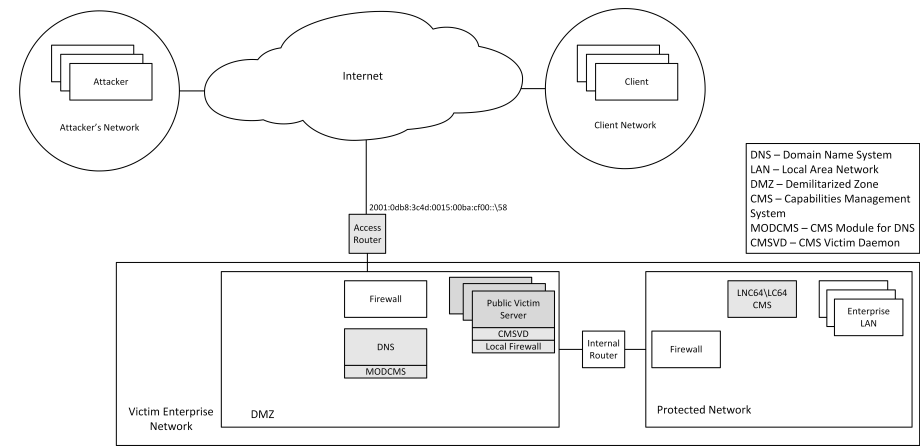


Figure 5.11: Logical diagram of the enterprise network showing components necessary to enhance client privacy.

Assumptions

We assume that ILNP locator updates are protected so that an eavesdropper is unable to extract the next active locator value. Otherwise, the eavesdropper can reconstruct a single session by following the locator updates. It is also assumed that the application data in a single client session is protected by encryption.

Emulation Configuration

We used a Python⁷ script for this emulation, and made the following two scenarios:

- Scenario one: Client to server communication without a LNC64/LC64 mechanism in place.
- Scenario two: Client to server communication with a LNC64/LC64 mechanism in place.

Both scenarios were emulated with 2 hours of data represented by 7,200 data points. We used only one subnet with scenario one but used eight subnets with scenario two. Randomness was introduced to the duration (i.e., the lifetime of an active subnet). We introduced randomness in active duration of a capability because it help us show the unpredictability of an active lifespan of a future LNC64 capability. This way an attacker does not even assess how long the next capability will be active for. It is an value added mechanism which is not a requirement. In all of our previous experiments with locator based defences, we used a fixed duration of either 10 seconds or 20 seconds. A maximum duration of 30 seconds is allowed for a particular subnet to be active in our evaluation. An example *subnet to lifetime* sequence is given in Table 5.3 for a part of a client session.

Locator (L64)	L1 (38)	L2 (19)	L3 (2b)	L4 (3c)	L5 (0d)	L6 (16)	L7 (36)	L8 (3f)
Active Duration (seconds)	6	25	30	12	1	6	11	19

Table 5.3: An example showing the active lifetimes of 8 locators in a chunk of a single client session.

5.3.3 Results

Figure 5.12 shows the results after running the emulation for 7,200 seconds.

⁷<https://www.python.org/>

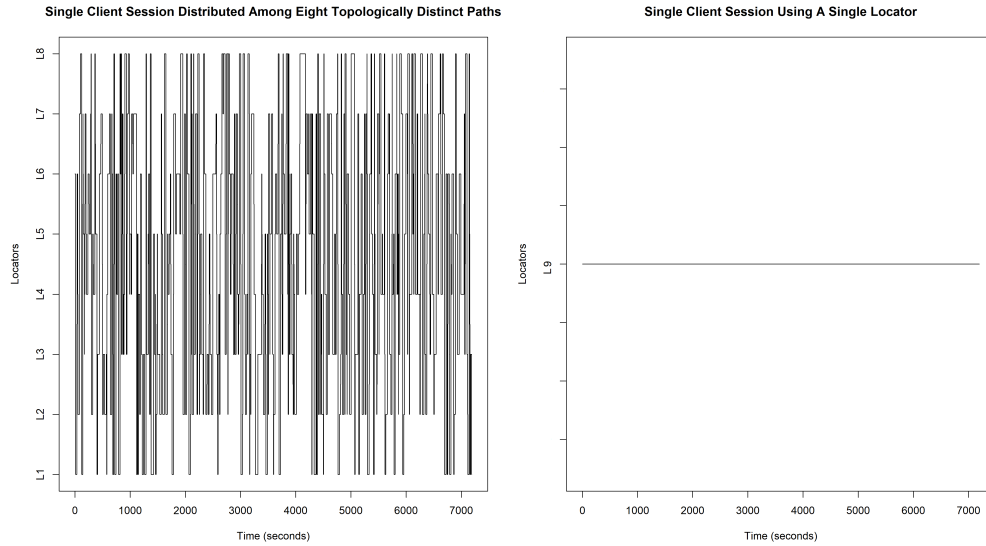


Figure 5.12: The results for the emulation showing the client privacy side effect of an LNC64/LC64 DoS defence. The scenarios one and two are shown at the left and the right side respectively.

If an eavesdropper looks at a client session generated by scenario two, then it would be difficult for it to trace the client session. As the session will use the same destination NID64 value, the eavesdropper will eventually reconstruct the session. If a session contains multiple TCP connections with each connection using a different NID64 value, then it would become harder for the eavesdropper to trace the session.

5.4 Summary

This chapter presented a multi-layered defence which can simultaneously protect an enterprise host and its network from low-rate TCP SYN flooding attacks and traffic-based volumetric DoS attacks (see §5.2). It combines the NC64 and the LC64 approaches, and provides a per-client NC64 capability that has multiple LC64 capabilities within its ILNP session (see §5.1). We termed these capabilities as L64 and NID64-based capabilities (LNC64).

The LNC64 defence was empirically tested and it was proved that it is possible to provide multi-layered enterprise security using both namespaces of the ILNP protocol (see §§5.2.1 and 5.2.2).

We also noticed a side-effect of this defence where there is a possibility of enhancing the privacy of a client session using the LNC64 defence (see §5.3). The basic requirement of such a privacy enhancement is to encrypt or secure ILNP control messages for mobility (see §5.3.2). To test the evidence of such a possibility, we emulated our defence for a single client session. The

emulation showed an ILNP client session distributed over multiple destination namespaces after enabling the LNC64 defence (see §5.3.3). Further rigorous analytical and empirical studies are required to explore the privacy features that can be provisioned with an ILNP protocol.

Chapter 6

Conclusion And Future Works

6.1 Summary And Contributions

Denial of Service (DoS) is one of the most challenging security threats to enterprises (see §2.3). It is easy to launch but hard to mitigate. This work focused on its mitigation, specifically for those attacks which use spoofed sources. By providing mitigations, this research also made it difficult for an attacker to launch an attack.

This work makes use of ILNP mobility which has been extensively tested in earlier researches [Phoomikiattisak & Bhatti, 2015], [Phoomikiattisak, 2016]. However the use of ILNP namespaces through Domain Name System (DNS) to mitigate DoS attacks has not been tested in earlier research. This work used ILNP mobility (see §2.1.3), its DNS resource records (see §2.2.3), along with new defence paradigms (see §§2.4.1 and 2.4.2). DNS capabilities and Moving Target Defence (MTD) paradigms have not been used prior to this research with ILNP as well.

The following are the major architectural and engineering contributions addressing our three research questions (see §1.7.2):

6.1.1 New Enterprise Security Architecture

A novel approach to security was designed to show how a MTD paradigm, a DNS fast flux mechanism, and the concept of DNS capabilities can be used with an ILNP mobility feature to mitigate low-rate SYN flood attacks (see §3.1) and bandwidth-exhaustive volumetric DoS attacks (see §§4.1 and 5.1).

ILNP has not been used before with DNS to secure enterprise networks. Similarly, the benefits provided by MTD, DNS capabilities, DNS fast flux, and ILNP mobility were used to form a security architecture that inherits advantages from these domains. An enterprise does not need to trust an

external body since all novel defences, provided through this research, work within an enterprise environment.

These defences also provide some level of flexibility in their implementation. An enterprise can use any protocol for internal control messages among defence backend entities, or it can colocate or disperse different elements of the defence at strategic locations to enhance performance or other relevant metrics.

These defences are our main architectural contributions.

6.1.2 Defence Provisioning For Enterprise Hosts Through The NC64 Defence

Chapter 3 provided a rationale for securing enterprise hosts against SYN flood attacks through ILNP's NID64 namespace-based DNS capabilities. It was proven that these capabilities provide a valid proof of concept (see §3.3) and eliminate shortcomings (see §3.1) of an alternate security mechanism, i.e., SYN Cookies (see §2.3.6.1).

These capabilities provide per-client access control which is beneficial for future traffic engineering for security as well. These DNS capabilities make use of host-based firewall and provide flexibility of aggregating as much client information as possible to grant capabilities through the backend (see §§3.2.2 and 3.2.3).

The NC64 defence has few important architectural and engineering limitations. It requires end systems to be ILNP-capable due to use of the NID64 namespace. It also requires a modification of the host firewall whenever a new capability is generated. DNS write performance is another engineering issue that can be solved through better engineering of CMS, MODCMS, and CMSVD.

It should be noted that this form of defence is possible through IPv6 as well (see §2.4.2.1). So, the NC64-based defence is an alternative approach to host security.

6.1.3 Defence Provisioning For Enterprise Networks Through LC64 Defence

Chapter 4 provided a rationale for securing an enterprise network as a whole against volumetric DoS attacks through ILNP's L64 namespace-based DNS capabilities. It was proven that these capabilities not only provide proof of concept (see §4.2) but they achieve security with minimal overhead (see §4.2.4). Similarly, this form of defence is not possible through native features of IPv6 (see §4.1).

Volumetric DoS attacks can render all enterprise services unavailable. Even though the enterprise services within its infrastructure might be running without any issues, to the outside world it will look as if the enterprise

is unable to serve external service requests due to its internal infrastructural issues. Our research has provided a state-of-the-art mechanism to shift an entire enterprise network to redundant uplinks with minimal overhead while maintaining existing transport layer connections. We noticed that the LC64 defence increased the client bandwidth (during an attack) once the solution was in place (see §4.2.4).

The LC64 defence has few important architectural and engineering limitations (see §4.1.1). It requires the end systems to be ILNP-capable. It also requires reconfiguration of access router interfaces whenever a new capability is generated. As an enterprise has full control of its access routers, we believe it can ease the deployment in future networks while improving our defences' engineering. DNS write performance is another engineering issue that can be solved through better engineering of CMS, and MODCMS.

6.1.4 Multi-Level Enterprise Defence Provisioning Through LNC64 Defence

Chapter 5 provided the rationale for simultaneously securing both enterprise hosts and their networks from variable-rate DoS attacks. We employed L64 and NID64 ILNP namespaces through a novel engineering of CMS (see §5.2) and its interaction with access routers and enterprise hosts. An enterprise benefits from security against low-rate SYN flood attacks, and bandwidth-exhaustive SYN and UDP flood attacks (see §§5.2 and 5.2.2).

The current security landscape demands multi-level security where multiple defence mechanisms are used to defend against a wide variety of DoS attacks that can occur simultaneously. Our defences are non-intrusive to other defences, i.e., they do not affect any other defence which can be simultaneously used by an enterprise.

The LNC64 defence shared the same architectural and engineering limitations introduced by NC64 and LC64 defences. Also the LNC64 type of defence is not possible through native functionalities of IPv6 (see §4.1).

6.1.5 Performance Measurements Of NC64 Defence In Diverse Environments

NC64 defence which is used against SYN flood attacks might behave differently in multiple delay and packet loss conditions. To test it against SYN Cookies under different network conditions, we ran multiple sub-experiments. We noticed that NC64 gave a valid proof of concept in a wide variety of network conditions (see sections 3.3.2.1 to 3.3.2.3 and 3.3.3).

We also noticed that as the network delay increases, the performance of CMS also increases. This was partly due to the fact that CMS performs better while it is not being fully loaded with capability requests. This helped

us decide that, if we optimise CMS performance, then we can increase the performance of our defence.

The measurements and insights obtained through this empirical evaluation enabled us to deliver it as our engineering contribution.

6.1.6 Performance Evaluation Of End-To-End Capability Distribution

§3.4 provided an empirical evaluation of the performance of the end-to-end DNS request/response mechanism in the presence and absence of NC64 capabilities. We tested capability distribution performance in multiple delay and packet loss environments. This evaluation helped us in understanding the performance difference in the presence and absence of our defence (see §3.4.3). We saw minute differences in performance but, with better engineering of the defence backend, we can achieve similar, if not better, performance.

This capability distribution comparison made an engineering contribution as through these measurements, we are able to assess the extra network latency that one can expect with our defences.

6.1.7 Eliminating The Capability Sharing Problem Through NC64 Defence

Capability sharing is the process in which a valid capability is shared with other clients (also see §2.4.1.2 from attacker's perspective). An attacker can take a valid capability and can distribute it among botnet machines to launch an attack. As the attacker-provided capability is valid, an enterprise host will allow the attacker's connection request.

To curb the above situation, we introduced a per-client capability using an enterprise host firewall and MAPs (see appendix A). A MAP can contain client naming information, current service information, e.g., port numbers, originating client network information, etc. We recommend using as much information about a client and an ongoing connection as possible (see §3.2.2). Based on this MAP, the firewall can decide whether to allow a client to access host services or not.

Since our approach is to have flexible content within the host MAPs, we introduce a powerful mechanism to counter the capability sharing problem. It is an important architectural contribution as capability sharing is an architectural problem in multiple non-ILNP based capability solutions.

6.1.8 Quantifying LU Overhead In LC64/LNC64 Defences

A locator overhead is defined as the number of locator update (re)transmissions required to shift a client session from one network to another. A Locator Update (LU) packet is a one-way packet that informs a client about new

active LC64/LNC64 values. An LU ACK is a single one-way packet that is sent from a client informing the server that its local ILNP bindings (see §2.1.2.1) are updated and hence data communications can proceed.

We made an average number of locator updates required for a single LU update to be successful in our testbed for LC64 defence (see §4.2.4). We also measured LU performance in terms of average Round Trip Time for such updates. We noticed an average of one locator update overhead for 60 handoffs in a 20 minute window. Similarly, for RTT measurements, we noticed an average of ~ 0.6 ms delay for each end-to-end LU (see Figure 4.8).

These measurements are important because we want to minimize the time and the number of locator update retransmissions to shift clients from the attacked path to the redundant paths. These measurements and assessments contribute as engineering contributions.

6.1.9 Crisp Separation Of Backend Control Traffic And End-To-End Data Traffic

NC64, LC64, and LNC64 defences have a similar backend (see §3.2.1, §4.2.2, and Figure 5.4) that utilizes concepts from the MTD paradigm, DNS capabilities, and DNS fast flux. Each control signal generated within the backend does not interfere with data traffic (see §§3.2.6 and 4.2.1). We mandated that each enterprise host and access router should have a dedicated link with the CMS. Similarly, we mandated that the link between CMS and MODCMS should be separate.

We used this separation in our testing and made a successful proof of concept for each defence (see §§3.3.1, 4.2.3 and 5.2.1). This separation is important for the security of the backend as well. This separation can be achieved in real networks by allowing control traffic to flow only within the internal network. This is effectively an architectural contribution.

6.1.10 Measuring Leaked Attack Traffic During Network Transitions In LC64 Defence

As LC64 defence requires mobility (see §2.1.3), it is important to measure attack traffic during a transition from one network to another. If attack traffic leaks to the new network then it can affect services, albeit for a short duration.

We measured attack traffic during the transitions and noticed that transitions were abrupt (thanks to low LU overhead (see §4.2.4)), and we found no attack traffic leakage (see §4.4.3). This measurement is effectively an engineering contribution.

6.1.11 Increasing Attacker's Effort

Planning an attack is an important phase for any successful DoS attack (see §§2.3.3 and 4.3). If a defence can increase this effort so that an attacker is discouraged from launching a successful attack, then it can contribute to further enterprise acceptance.

NC64 defence introduces a per-client capability concept (see §3.2). If an unauthorized client scans a host (e.g., for open ports), where NC64 based capabilities are active, then it will not receive any packet replies conveying information about open ports etc. If an attacker is trying to test vulnerabilities in an enterprise host, then it would be difficult for it to assess vulnerabilities because its packets cannot reach the Transport or Application layer of the enterprise host.

LC64 defence introduces the concept of short-term capabilities that are fast fluxing end-to-end traffic among diverse Internet paths, i.e., the defence uses a different uplink path for each capability (see §4.1.1). If an attacker has launched an attack on one path then, after a short duration, it has to launch the same attack on a different path (for which it has to prepare again since the next path is unpredictable). Since an attacker has to follow each path or it has to attack all the paths in order to cripple enterprise service availability, this will increase the effort on the attacker's side.

LNC64 defence combines NC64 and LC64 modes (for engineering details see §5.2). It would therefore be difficult for an attacker to plan and then launch a successful attack in a LNC64-enabled enterprise network.

The above mentioned aspects of NC64, LC64, and LNC64 contribute architectural contributions in the context of an increased effort of an attacker.

6.1.12 Side Effect: Possibility To Enhance Client Privacy

Chapter 5 presents a defence mechanism in which a single client session can take multiple paths within its lifetime. Each different path means that the session has different destination namespaces. ILNP achieves this in-session mobility through a locator update mechanism. Locator updates convey new network information to the clients. If we can provide encryption (or any other form of protection) to these locator update packets, then an eavesdropper will not be able to assess whether the client is using a single session or multiple sessions with different flows (see §5.3.2).

LC64/LNC64 mechanism-based privacy can be used as an alternative to MPTCP ADD_ADDR-based similar mechanisms (if ADD_ADDR TCP option itself is protected too) (see §5.3.1).

We provide an emulation in §5.3 to demonstrate this side effect. This side-effect is an architectural contribution with an architectural limitation of having encrypted ILNP-based locator updates.

6.2 Discussion

6.2.1 Security Of ILNP-Based DNS Capability Defences

Figure 6.1 shows a logical enterprise diagram with all new architectural components required for our defences.

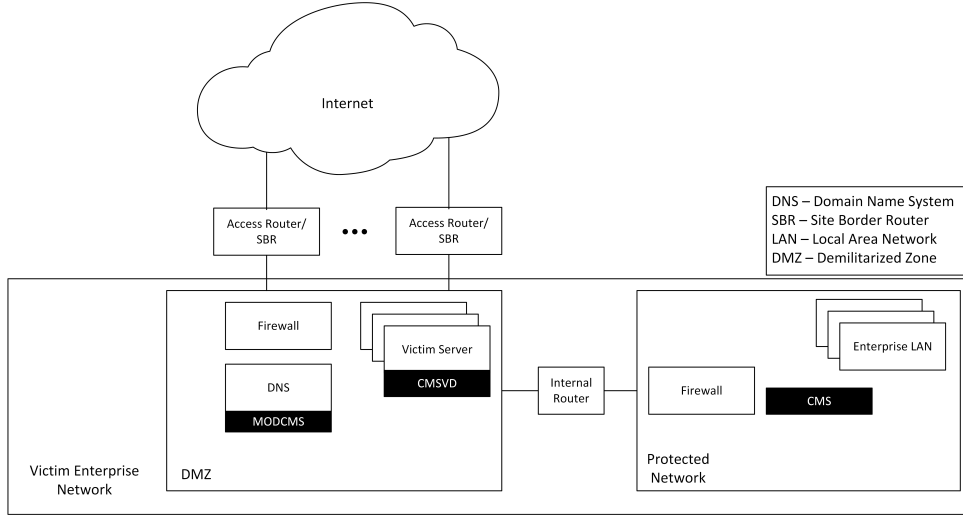


Figure 6.1: Logical diagram of an enterprise network. Each new required element for defences is shown in black.

Each presented ILNP-based defence has a backend which is composed of a defence-specific CMS, a CMSVD, a MODCMS, and a DNS. The CMS is part of the protected network, i.e., it is behind a Demilitarized Zone (DMZ). The identity of the CMS is secret to an enterprise. The CMSVD daemon runs on a publicly accessible enterprise host in the DMZ, and it is aware of the location of the CMS. The DNS server is deployed in the DMZ. The MODCMS is collocated with the DNS and knows the location of the CMS. Each access router which participates in the defences can be connected to the CMS.

Each connection among components within the enterprise backend are secured by TLS. An outside eavesdropper cannot get the next value of a capability since its creation and configuration is done within the enterprise.

As location and identity of the CMS is hidden, it is not possible for an attacker to make a DoS attack on the capability orchestration backend. For capability distribution, an enterprise uses the same uplinks so an attacker can congest the uplinks to launch a successful denial of service attack. If LC64 and LNC64 defences are enabled, then it will not be possible for an attacker to launch a successful denial of capabilities attack, unless it makes more effort to congest all upstream links.

If a NC64 defence is active, then any volumetric DoS attack will make it ineffective because a NC64 defence is only suitable for spoofing-based low-rate TCP SYN flood attacks. If an attacker uses a TCP SYN flood using a client which has a valid capability, then it can cripple the NC64 defence. It is recommended that a strong mechanism should be in place to distinguish good traffic and bad traffic. There should be a mechanism to assess the behaviour of an authorized client, so that the CMS can later revoke a capability from a misbehaving client.

If a LC64 defence is active and a low-rate TCP SYN flood is launched on all uplinks of an enterprise, then it can cripple the establishment of incoming TCP client connections. To curb this situation, we recommend that an enterprise should use either a LNC64 defence, or it should use SYN Cookies/Caches (or similar) along with a LC64 defence.

If a LNC64 defence is active, then limited availability of enterprise uplinks or a capability of an attacker to launch an attack which can congest all enterprise uplinks, will become a bottleneck. This bottleneck can be managed by provisioning of more uplinks and performing measurement to continuously check the effectiveness of the solutions.

If an attacker attacks the DNS with a volumetric DoS attack, then all defences will be negatively impacted as this will constitute a denial of capabilities attack. To mitigate such an attack, an enterprise can have more than one entry point for DNS name resolution. The threat of denial of capabilities should be further investigated in future research.

CMSVD runs in the publicly accessible enterprise host. It is not possible to directly attack the CMSVD since it has a dedicated link to the CMS. It has a host firewall dependency. CMSVD can only be attacked successfully if an attacker has control (*root* user role) of the host operating system.

6.2.2 Resource Consumption Of CMSVD, MODCMS, CMS, And Router Scripts

The CMSVD daemon, which is used by the NC64 and LNC64 defences, uses one TCP connection to communicate with the CMS. It uses less than ~0.1% of system memory and CPU at the victim host. For CMSVD, `nftables`¹ modules must be enabled within the kernel. It also requires root access to the Linux system, since the firewall rule installation is dependent on root user. So CMSVD is protected from any non-root user who might compromise the host to manipulate firewall rules.

For NC64 defence, the CMSVD configures a single interface with the total number of capabilities that CMS creates. It synchronizes its values using a seed provided by CMS. If CMS allocates 1,000 capabilities then CMSVD will configure 1,000 secondary interface addresses/names before

¹<https://netfilter.org/projects/nftables/>

any DNS request comes through CMS. For LNC64 defence, CMSVD will configure one or more interfaces based on the requirements and it would have to use all the NID64 values for each of the L64 values.

The MODCMS daemon, which is collocated with DNS, uses one TCP connection to communicate with the CMS. In our implementation, this is a part of the KnotDNS². It is only limited by the maximum amount of memory that is used by KnotDNS software. It has a minimal CPU overhead due to its limited responsibilities. For each capability request from the client, the path from it to the CMS uses at most two messages (a capability request and a capability response).

The CMS daemon, which is a standalone software that runs on the CMS server, uses one TCP connection to communicate with the victim host, one TCP connection to communicate with MODCMS, and one or more SSH connections to run scripts on the upstream routers. The CMS software consumes less than $\sim 0.1\%$ of system memory and CPU.

In LC64 and LNC64 defences, each participating upstream router has a script that updates the radvd's³ configuration file, reloads the radvd daemon, and adds the iptables⁴ rules for new routes. It takes less than $\sim 0.1\%$ memory and CPU to execute once.

6.2.3 Short Duration DoS Attacks

Attackers, sometimes, use short duration DoS attacks to thwart attack detection, its mitigation, and attack correlation. All ILNP defences presented in this work can equally work for mitigation of such attacks. As these defences cannot be used for detection and analysis, so it is a requirement that an enterprise should deploy attack detection mechanisms for such attacks as well. We recommend an automated mechanism to signal the start of these defences, as they can be enabled and disabled instantaneously.

6.2.4 DoS Attacks From Within The Enterprise

If a TCP SYN flood is launched from within the enterprise then it is possible to mitigate these attacks if internal clients also get capabilities from CMS through the internal DNS. If it is a volumetric attack then an enterprise needs multiple links at the sub-network edges. This research only dealt with external attacks, but it is possible to use a similar solution within an enterprise.

²<https://www.knot-dns.cz>

³<http://www.litech.org/radvd/>

⁴<http://ipset.netfilter.org/iptables.man.html>

6.2.5 Distributing Traffic Load Among Upstream Providers In LC64/LNC64

For LC64/LNC64, a DoS attacker has to attack on all available upstream providers for a successful attack. If there are two 10 Gbps links to the Internet, then the attacker has to have 20 Gbps capacity to attack availability on both links. Each participating upstream router must be connected to CMS. The only limitation for CMS scalability is the number of TCP connections it can support to/from routers, DNS (or MODCMS module) servers, and victim hosts.

Normally, an organization is connected to the Internet through more than one link. If an organization is provided with one /56 prefix/L_pp then the LC64 defence is ineffective. This does not hold true for the NC64 defence which primarily deals with low-rate DoS attacks.

6.2.6 Fast-flux Multi-homing

When the victim server is connected to multiple upstream providers, it is known to be multi-homed. In the case when a single TCP connection can use multiple upstream providers for its communication, then this is termed as multi-path communication.

In LC64 and LNC64 defences, victim servers will be connected to one particular upstream provider at one time. So traffic coming on all other non-active upstream providers will be blocked. The victim server will be continuously changing its upstream providers and this will form a solution that can be called a fast-flux mobility for defence. As our evaluations did not use two or more concurrently-active upstream links, we cannot say, unless empirically tested, that our defences are applicable to fast-flux multi-homing. It can be empirically tested that if a CMS makes two links active at the same time so that ILNP-specific multi-homing feature can be used on both the active links. Later on, this mechanism can employ the ILNP-specific mobility feature to concurrently shift to two different upstream links.

If we enable soft handoff during a transition from one upstream link to another in LC64 or LNC64 defence, then each victim will be effectively multi-homed during the transition.

6.2.7 Vulnerability Scanning And Penetration Testing

NC64, LC64, and LNC64 defences will help reduce vulnerability testing, node/site probing, and scanning (for definitions see §2.3.3). Ideally, LC64 and LNC64 defences should rotate capabilities in such a way that by the time an attacker starts a successful scan, the server should have moved to a topologically significant path. Similarly, in NC64 defence, an enterprise should make sure that it identifies miss-behaving clients with capabilities.

Using LNC64/LC64, it would become difficult for the security assessment team of an enterprise itself to do penetration testing. We would recommend that an enterprise should test different LNC64/LC64 configurations (e.g., duration of fast-flux, using particular upstream links, etc.) on a small enterprise network representative testbed to assess which particular configuration will be best for deployment. If there is an assessor (e.g., an assessment specialist) to assess how much effort the attacker would need, then the level or frequency of fast-flux could be defined based on this information.

6.2.8 ILNP Namespace Spoofing

ILNP requires a nonce header (at least) at the start of any TCP or UDP transport session. A nonce header contains a nonce value which is unique for each transport session. It is also possible to have a nonce value in every ILNP packet. The nonce header uses nonce values which are part of the ILCC state for that transport session.

If an off-path attacker uses spoofed source namespaces for an attack, then our solutions will easily mitigate (based on the native features of ILNP) that attack since the machine whose address has been spoofed will not be able to confirm the nonce from initial setup of communication. In ILNP, nonce values are part of the ILCC bindings for a particular communication session. For initial setup of communications, if a client does not have a nonce for a source namespace, then it drops the packet. Figure 6.2 shows this scenario.

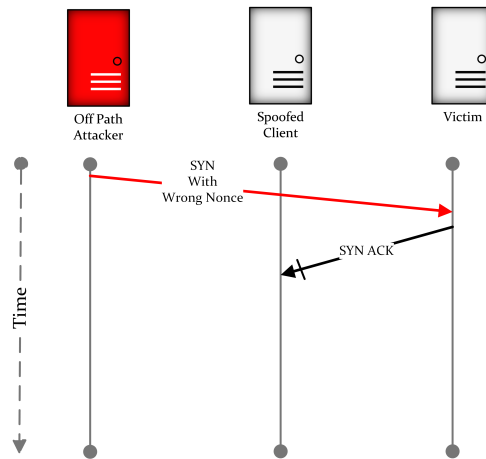


Figure 6.2: ILNP namespace spoofing and its effects on TCP connection establishment as done by an off-path attacker.

If an on-path attacker uses spoofed source namespaces for an attack, then our solutions will not be able to mitigate (based on the native features of ILNP) that attack since the machine whose address has been spoofed will be able to confirm nonce through its ILCC bindings. For the initial

setup of communications, if a client has a nonce for a source namespace then it accepts the packet. For these reasons, we recommend to use as much information as possible in the MAPs specific to NC64 and LNC64 defences. In case of L64-based defences, an attacker has to be on all the paths to compromise the defence. Figure 6.3 shows this scenario.

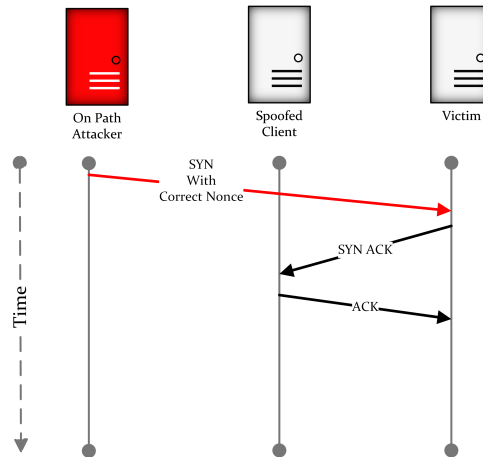


Figure 6.3: ILNP namespace spoofing and its effects on TCP connection establishment as done by an on-path attacker.

In the case of IPv6 capabilities, since the nonce is not required for the initial setup, an attacker would be able to create an initial connection to the victim server. But as the victim server will fast flux using another prefix, the machine whose address has been spoofed will not be able to send an acknowledgement to the binding update. Mobile prefix discovery in binding updates to the correspondent node uses the return routability procedures. The machine whose address has been spoofed will not be able to participate in the binding update return routability procedure. Hence, the connection would be nulled automatically.

6.2.9 Using Botnets For An Attack

An attacker can use an automated attack program called bot on a compromised client. If a client on which the bot is running is an authorized client, then it will affect communications even if the defences are in place. If an attacker is using bots for a volumetric DoS attack, then the bots will see locator updates during execution of DNS fast-fluxing using LC64/LNC64 defences. Then, bots will be able to follow the fast fluxing, and LC64 and LNC64 defences will become ineffective. We recommend that an enterprise should deploy alternate solutions against attacks which are launched using bots.

Some attacker-owned scripts, i.e., viruses, can go undetected in a client's

anti-virus software. For such a scenario, our defences are ineffective. If an enterprise sees a successful attack using low-rate SYN flooding, then it needs to investigate its current list of authorized clients and take appropriate action.

6.2.10 Displacement Of Control From DNS To CMS

Solutions provided in this research do not rely on on-path gateways which, mostly, conflate connectivity, control, and security. We made sure that the Application Programming Interface (API) between client and DNS remains the same while our solutions are in place. Complete control of capabilities management is behind the DNS, which provides a management layer in a protected enterprise environment. DNS is in the DMZ but CMS is in a protected network, i.e., behind the DMZ.

6.2.11 Defence Against Botnet Structure Creation

A continuous change in network configuration, e.g., change in NID64 and L64 values makes it difficult for a DoS command and control centres to form effective botnets. Each time there is a change in network configuration, a botnet has to be reconfigured, and in the face of fast fluxing, attackers have to cope with these changes. It is best practice for an attacker to minimize their contact with botnets, so the defences given in this research make it difficult for botnet owners to hide their identity.

6.2.12 Attack Planning, And Durable Information Aggregation

Although the objectives of this research do not contain an empirical investigation on whether an ILNP-based defence can stop an attacker's abilities to plan an attack or not, it can be analytically established that these defences provide obstacles for a DoS attacker in planning an attack. The planning phases of a successful attack initiation contains information gathering using reconnaissance attacks, mapping networks, e.g., using `nmap`⁵, port scanning, link monitoring, etc. Using an ILNP-based defence, attackers have to keep track of NID64 values and/or L64 values. Since these values are highly unpredictable to guess, it becomes a difficult job to keep track of them.

An attacker can do port scanning (using a bot) since we haven't included port number randomization in this research, but our shared mapping between the CMS and the victim has the ability to store any information (thanks to the latest provisioning in the netfilter module and nftables package). This feature demands future investigation.

⁵<https://nmap.org/>

6.2.13 End-to-end QoS

End-to-end QoS measurements in the context of a client-side evaluation is an important aspect for any enterprise. Clients are concerned about performance and availability of services and networks, whereas DoS or network congestion negatively affects both of these aspects. In this work, we measured availability using bandwidth measurements at the server side using `iperf3`'s TCP stream. There are multiple ways to measure QoS for clients, e.g., through ICMPv6, TCP, HTTP, etc. We rejected the use of ICMPv6 for measuring availability because of its unreliability in determining the service effects running on enterprise networks [Z. Hu et al., 2014]. So we needed a better end-to-end monitoring protocol than ICMPv6. We chose TCP as it gives us information about the end-to-end connection state which is a better predictor of availability.

Another aspect of QoS is service performance, which can be measured using TCP capture at the client side, server side, or a capture device which sits in between the client and the server.

6.2.14 Industrializing The CMS-based Backend And Scalability

This work is based on ILNPv6-specific and thoroughly-tested Linux kernel features. If an ILNPv6 implementation is provisioned within the current production enterprise environments, it would be possible to implement, test, and provision security through our solutions.

The CMS backend of our solutions can be scaled-up/down or scaled-in/out since it works in standalone mode. If the enterprise wants to run CMS backend in a distributed environment with state sharing, then a further study would be required to check solutions' effective or performance.

6.2.15 Embedding Intrusion Detection System (IDS) In CMS Backend

IDS can enhance the security provisioned by our backend. An IDS can create ACLs which can contain black-listed source addresses/namespaces or bad-behaving clients which have active and valid DNS capabilities. The CMS backend can then signal the DNS to restrict forwarding of capability requests to CMS.

Similarly, an IDS can create specialized white-lists of source address/-namespaces which indicate to CMS that it can grant DNS capabilities to these clients for an extended duration, thereby reducing load on the CMS backend and providing extra capacity to the capability request/response channel.

6.3 Future Works

While this research provides a proof of concept for DoS mitigation through ILNP and DNS, there is more that can be done to solidify it before launching it in real networks. We list possible future works relevant to this research in the following:

- **Component/testbed Optimizations** — CMS, MODCMS, and CMSVD are custom-built software for this research. CMS and CMSVD support concurrency, but MODCMS does not. MODCMS is based on the modular architecture of KnotDNS. Similarly, all of these components can be further optimized to maximize performance. The main requirement for CMS is to have a hidden location and identity from outside of the enterprise. If it is possible to colocate it with DNS (which is in the DMZ), while maintaining its privacy, then latency can further be reduced. Furthermore, if the CMS can provide hopping sequences and timing information to all routers for an extended period of time, then the latency can further be reduced.
- **Testing with other DoS attacks** — We tested ILNP-based DoS defences with low-rate TCP SYN flooding attacks and TCP/UDP flooding attacks. Conceptually, it can be used or enhanced to mitigate other forms of DoS attacks. It requires testing with spoofing-based Application layer attacks as well.
- **Investigations with botnets** — The focus of our research was spoofing based DoS attacks. There should be investigation on the possibility of mitigation of botnet-based DoS attacks through ILNP-based defences.
- **Viability for SDN systems** — SDN systems are heavily used in enterprises, and DoS attacks are also being launched on SDN networks. Future work demands an investigation into the viability of DoS mitigation through ILNP defences in SDN networks.
- **Testing with fast-flux multi-homing** — In LC64 and LNC64 defences, the victim host/network shifted from one upstream link to another through the use of mobility. If there is a requirement to use more than one upstream providers simultaneously, then a host/network can also be shifted to two or more uplinks simultaneously to maintain the requirement. This scenario can be tested by modifying CMS.
- **Evaluating and mitigating backend security** — Different backend security issues have been outlined in §6.2.1. These issues must be further investigated to solidify the integrity of ILNP-based defences.
- **Evaluating client privacy enhancements** — We presented emulation of the possibility of client privacy enhancements which is a side effect

of our defences (see §5.3). It should be investigated by performing empirical evaluation of this side-effect.

- Investigating DoS attacks from within the enterprise — We addressed enterprise security from external DoS attacks. It would be beneficial if the same can be evaluated if the attacks originate from inside which not only target normal enterprise hosts but also CMS and internal DNS.
- Evaluation in real networks — We tested our solutions in a laboratory network with emulation of different network conditions. It would be beneficial if the same can be evaluated in real enterprise networks with real network conditions. Both results can be compared and relevant further research can be performed.
- CMS MAP's *context enhancements* — NC64 and LNC64 defences require a mapping between the client information and ephemeral capabilities to be installed in the victim host firewall. This research used a MAP that contains client's NID64 and L64 values mapped to ephemeral capabilities. If an attacker can spoof authorized client's NID64 and L64 values, then NC64/LNC64 defence can become ineffective. To curb such a situation, we can introduce further granularity in client information within the MAP. Such information might be source/destination port numbers for the connection and the source network information of the client, etc.

Appendix A

Identifier-based Mappings' Specification

In response to a DNS query, CMS creates a unique mapping to be configured at a victim host. There are two parts to this configuration. The first part configures victim's network interfaces with all the Node IDentity (NID) values along with a fixed L64 value for each. The second part of the configuration is the configuration of the firewall in the victim. The first part has to have taken place beforehand while the second part of the configuration is dynamic and it only takes place when a new capability request comes in to CMS.

Each mapping contains data which is used by a victim to make decisions about resource access authorization. There is a single type of MAP which will show information about active capabilities. A MAP is stored at a CMS and a victim. It is the responsibility of a victim to enable filtering, based on the information provided in the MAP. If there is a MAP conflict or a victim is unable to update its MAP then it has to inform the CMS using an error condition.

MAP is communicated between a CMS and a victim through a Colfer¹ based binary serialization format protocol. Whereas, control plane communication has two parts. The first part deals with a MAP advertisement which goes from the CMS to the victim. And the second part is a response from the victim containing MAP acknowledgement message or error condition to the CMS. If the CMS does not receive an ACK for MAP advertisement then it will discard this particular MAP entry from its table. And, in response to the capability request from the MODCMS, it generates an error which effectively makes the DNS generate a NXDOMAIN message back to the client so it can make a fresh request.

¹<https://github.com/pascaldekloe/colfer/wiki/Spec>

A.1 Control Message Types

Each control message contains an *msgtype* field which uniquely identifies the type of message in transit. Table A.1 presents all the *msgtypes*:

Table A.1: Message types used in an Identifier-based capability system

Message Type (<i>msgtype</i>)	Code	Meaning
MSGTYPE_MAP_PUSH	0	MAP message from CMS to victim server
MSGTYPE_MAP_ACK	1	MAP Acknowledgement from victim to CMS
MSGTYPE_SEED_POLICY	2	Seed information from CMS to target which helps victim to generate NID values which are exactly the same as in the CMS software.
MSGTYPE_SEED_POLICY_ACK	3	Seed information Acknowledgement from target to CMS
MSGTYPE_CAP_REQUEST	4	Capability request message from DNS to CMS containing client information e.g. client address, originating network, service to be requested etc.
MSGTYPE_CAP_RESPONSE	5	Capability response message from CMS to DNS

Control messages are transported over a TLS based encrypted channel, which secures these control messages even though these are within an enterprise network. But the management can decide to use plain-text messages for throughput or performance reasons.

A.2 ILNPv6 Specific MAP Advertisement

A MAP advertisement contains a client's NID and L64 values; a victim's one or more ephemeral NID values; and one or more fixed L64 values, as shown below in the colfer based data structure:


```

type map struct {
msgtype  uint32,
mapid    uint32,
C_NID64  text,
C_L64    text,
E_NID64_1 text,
E_NID64_2 text,
...,
L64_1    text,
L64_2    text,
...
}

```

For experimental and proof of concept purposes, one capability per message is used in this research.

A.3 Map Acknowledgements (MAP ACKs)

A victim will need to send a MAP acknowledgement upon receipt of a MAP advertisement. MAP acknowledgement is an explicit message that contains a MAP ACK along with a MAP_ID.

A.4 Map IDs And Capability IDs

Each message between a CMS and a victim either contains a MAP, an ACK, or an ERROR based on a specific situation. Each MAP also contain a MAP ID. This information might be used for the capabilities' traffic engineering mechanisms e.g. in future one can define an explicit message from a victim, specifying a list of clients who should be black-holed by a security enforcement agent. In that case, a victim only has to send a list of MAP IDs to any other system which has MAP ID to client information in its configuration.

Similarly a Capability (CAP) ID is used between a DNS and a CMS. A database in a CMS maintains information about which CAP ID is connected to which MAP ID and hence the mapping. A numerical counter is controlled at the DNS for CAP IDs. A mapping can also contain information about originating DNS server of the capability request message so CMS can maintain another list containing information about which CAP IDs belong to which originating DNS servers. A CAP ID ensures that some of the state in the DNS is actually shared with CMS, which can decide future responses for certain queries.

A.5 MAP Expirations

Expiration of a mapping means that a specific client should be allocated a new capability in the next DNS request. A victim will keep communicating with the client in the same session using old capability but for any other new session a new capability will be generated.

In our testbeds, we have set an expiration TTL to zero which mean a specific client will always get a new capability whenever it asks for a DNS resolution. The TTL for the DNS records will be zero, so, whenever a client wants to communicate with a victim, it has to send a DNS resolution request. This way, the client will be obliged to access the victim's new identity from DNS server i.e. automatically complying with the management interface of the capabilities distribution mechanism.

Mapping's TTL is a duration of the mapping before it expires. It has an indirect correspondence with the DNS record's TTL. It is up to the victim to define a policy which dictates the Mapping's TTL, hence DNS TTL. One possible policy might be 5 seconds TTL for all the current mappings. A victim is also able to send a MAP expiry message to a CMS if it needs to restrict policy for any client. This message specifically asks the CMS to expire the current mapping, but it also means that the victim wants to break any communication with the client on an old CAP. Upon reception of an expire message, it can generate another mapping after a new DNS query.

If a mapping is expired during an existing TCP connection, we will lose the session continuity objective but it is up to the victim to use the old capability until a new session with a new capability is initiated. Since control channel is secure so fake MAP expiry messages will be ineffective. Also, for reasons of connection continuity, we will postpone the TTL based map expirations to the locator-based solution. For an identifier-based solution, an expiry will mean breakage of the communication, and re-establishment of the connection unless victim wants to keep an old capability for current session (and rejecting any new session with an old capability).

A.6 Security Of MAP Related Control Messages

As mentioned earlier that communications among CMS, DNS, and victim are secured by a TLS protocol. Security of these control messages is not a hard requirement but it is up to the management to use any particular trusted security protocol. Since control message communication is within enterprise network, an enterprise can decide otherwise, for throughput or performance reasons. The victim's persistent connection to CMS is over a fixed address information which is within the CMS configuration. The address of CMS is hidden in the DNS server's configuration. This way, address information about CMS and victim is totally hidden from clients.

The client can initiate a data connection in whatever way possible as dictated by the end service. For our experimental purposes we have used NGINX² client/server HTTP application.

A.7 Firewall Considerations

A victim uses host-based firewall called nftables³. nftables provides new in-kernel packet classification framework. It also provides mechanisms through which we can build extremely fast firewall rules in the kernel. Our codebase contains C programming code which directly manipulates firewall rules using nftables provided Application Programming Interface (API). Mapping is created within the kernel which contains only packet *accept* rules. Other rules pertaining to *input* chains are in *deny* state. CMS address information is always configured as a trusted entity in the firewall.

A.8 Capability Format Considerations

Capabilities are created using a random seed and a secret key. The pre-shared secret key, between a CMS and a victim, is communicated at the start of communication. A configurable number of random seeds are sent to the victim from the CMS. The CMS and the target create unique ephemeral addresses using these seeds and the secret key. The victim then configures its interfaces with these addresses. This way it is ensured that there will not be any snooping since the capabilities are independently generated at victim and CMS.

²<https://www.nginx.com/>

³<https://netfilter.org/projects/nftables/>

Bibliography

- [Al-Dalky et al., 2018] Al-Dalky, R., Rabinovich, M., & Allman, M. (2018). Practical challenge-response for dns. *SIGCOMM Comput. Commun. Rev.*, 48(3), 20–28.
- [Al-Qudah et al., 2016] Al-Qudah, Z., Johnson, E., Rabinovich, M., & Spatscheck, O. (2016). Internet with transient destination-controlled addressing. *IEEE/ACM Transactions on Networking*, 24(2), 731–744.
- [An & Weber, 2016] An, N. & Weber, S. (2016). On the performance overhead tradeoff of distributed principal component analysis via data partitioning. In *2016 Annual Conference on Information Science and Systems (CISS)* (pp. 578–583).
- [Andersen, 2003] Andersen, D. G. (2003). Mayday: Distributed filtering for internet services. In *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, USITS'03 (pp. 3–3). Berkeley, CA, USA: USENIX Association.
- [Anderson et al., 2004] Anderson, T., Roscoe, T., & Wetherall, D. (2004). Preventing internet denial-of-service with capabilities. *SIGCOMM Comput. Commun. Rev.*, 34(1), 39–44.
- [Arbor Networks, 2010] Arbor Networks, A. (2010). 5th edition of the worldwide infrastructure security report. <https://goo.gl/3jyQuy>. Accessed: July 17, 2018.
- [Argyrazi & Cheriton, 2005a] Argyrazi, K. & Cheriton, D. (2005a). Network capabilities: The good, the bad and the ugly. In *HotNets-IV*: ACM.
- [Argyrazi & Cheriton, 2005b] Argyrazi, K. & Cheriton, D. R. (2005b). Active internet traffic filtering: Real-time response to denial-of-service attacks. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference, ATEC '05* (pp. 10–10). Berkeley, CA, USA: USENIX Association.
- [Arun & Selvakumar, 2009] Arun, R. K. P. & Selvakumar, S. (2009). Distributed denial-of-service (ddos) threat in collaborative environment - a

- survey on ddos attack tools and traceback mechanisms. In *2009 IEEE International Advance Computing Conference* (pp. 1275–1280).
- [Atkinson & Bhatti, 2012a] Atkinson, R. J. & Bhatti, S. N. (2012a). *Identifier-Locator Network Protocol (ILNP) Architectural Description*. RFC 6740, RFC Editor.
- [Atkinson & Bhatti, 2012b] Atkinson, R. J. & Bhatti, S. N. (2012b). *Identifier-Locator Network Protocol (ILNP) Engineering Considerations*. RFC 6741, RFC Editor.
- [Atkinson et al., 2009a] Atkinson, R. J., Bhatti, S. N., & Hailes, S. (2009a). Ilnp: Mobility, multi-homing, localised addressing and security through naming. *Telecommun. Syst.*, 42(3-4), 273–291.
- [Atkinson et al., 2009b] Atkinson, R. J., Bhatti, S. N., & Hailes, S. (2009b). Site-controlled secure multi-homing and traffic engineering for ip. In *MIL-COM 2009 - 2009 IEEE Military Communications Conference* (pp. 1–10).
- [Atkinson et al., 2012] Atkinson, R. J., Bhatti, S. N., & Rose, S. (2012). *DNS Resource Records for the Identifier-Locator Network Protocol (ILNP)*. RFC 6742, RFC Editor.
- [Atlasis, 2012] Atlasis, A. (2012). Attacking ipv6 implementation using fragmentation.
- [Bagnulo et al., 2015] Bagnulo, M., Paasch, C., Gont, F., Bonaventure, O., & Raiciu, C. (2015). *Analysis of Residual Threats and Possible Fixes for Multipath TCP (MPTCP)*. RFC 7430, RFC Editor.
- [Ballani et al., 2005] Ballani, H., Chawathe, Y., Ratnasamy, S., Roscoe, T., & Shenker, S. (2005). Off by default! <https://www.microsoft.com/en-us/research/publication/off-by-default/>. Accessed: May 11, 2016.
- [Bhardwaj et al., 2016] Bhardwaj, A., Subrahmanyam, G. V. B., Avasthi, V., Sastry, H., & Goundar, S. (2016). Ddos attacks, new ddos taxonomy and mitigation solutions — a survey. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)* (pp. 793–798).
- [Bhatti et al., 2016] Bhatti, S., Phoomikiattisak, D., & Simpson, B. (2016). Ip without ip addresses. In *Asian Internet Engineering Conference Bangkok, Thailand, November 30 - 2 December, 2016* United States: ACM. D. Phoomikiattisak was funded by the Thai Government. B. Simpson was funded by Cisco Systems under a University Research Programme (URP) grant award.

- [Bhatti & Atkinson, 2011] Bhatti, S. N. & Atkinson, R. J. (2011). Reducing dns caching. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 792–797).
- [Brisco, 1995] Brisco, T. (1995). *DNS Support for Load Balancing*. RFC 1794, RFC Editor.
- [Brownlee et al., 2001] Brownlee, N., Claffy, K. C., & Nemeth, E. (2001). Dns measurements at a root server. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 3 (pp. 1672–1676 vol.3).
- [Carnegie Mellon University, 1999] Carnegie Mellon University, C. (1999). 1999 cert incident notes. <https://goo.gl/wm3M4c>. Accessed: July 16, 2018.
- [Carpenter et al., 2010] Carpenter, B., Atkinson, R., & Flinck, H. (2010). *Renumbering Still Needs Work*. RFC 5887, RFC Editor.
- [Casado et al., 2006] Casado, M., Garfinkel, T., Akella, A., Freedman, M. J., Boneh, D., McKeown, N., & Shenker, S. (2006). Sane: A protection architecture for enterprise networks. In *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, USENIX-SS'06 Berkeley, CA, USA: USENIX Association.
- [CDNetworks, 2017] CDNetworks (2017). Q2 2017 DDoS attack trends report. https://www.cdnetworks.com/sg/resources/CDNetworks_DDoS%20Attack%20Trends_Q2%202017_ENG_final_20170821-2-.pdf. Accessed: July 16, 2018.
- [cert.org, 1996] cert.org (1996). Tcp syn flooding and ip spoofing attacks. https://resources.sei.cmu.edu/asset_files/WhitePaper/1996_019_001_496172.pdf. Accessed: July 16, 2018.
- [Cheng et al., 2016] Cheng, L., Divakaran, D. M., Lim, W. Y., & Thing, V. L. L. (2016). Opportunistic piggyback marking for ip traceback. *IEEE Transactions on Information Forensics and Security*, 11(2), 273–288.
- [Cloudflare, 2013] Cloudflare (2013). The ddos that almost broke the internet. <https://goo.gl/aeMZf9>. Accessed: July 17, 2018.
- [Cole, 2018] Cole, G. (2018). Syn cookies ate my dog - breaking tcp on linux. <https://kognitio.com/blog/syn-cookies-ate-my-dog-breaking-tcp-on-linux/>. Accessed: July 10, 2018.
- [Computer Weekly, 1992] Computer Weekly (1992). R. v. Richard Goulden.

- [Conklin et al., 2018] Conklin, W. A., White, G., Chuck, C., L. Davis, R., & Williams, D. (2018). *Principles of Computer Security: CompTIA Security+ and Beyond*. McGraw-Hill Education. 5th Edition.
- [D. J. Bernstein,] D. J. Bernstein. SYN cookies. <http://cr.yp.to/syncookies.html>. [Accessed: May 11, 2016].
- [Dierks, 1993] Dierks, M. P. (1993). Computer network abuse. *HARVARD Journal of Law and Technology*, 6, 307–342.
- [Dierks & Allen, 1999] Dierks, T. & Allen, C. (1999). *The TLS Protocol Version 1.0*. RFC 2246, RFC Editor. <http://www.rfc-editor.org/rfc/rfc2246.txt>.
- [Du & Abe, 2008] Du, P. & Abe, S. (2008). : (pp. 93 – 96).
- [Durcekova et al., 2012] Durcekova, V., Schwartz, L., & Shahmehri, N. (2012). Sophisticated denial of service attacks aimed at application layer. In *2012 ELEKTRO* (pp. 55–60).
- [Dyn, 2016] Dyn (2016). Dyn analysis summary of friday october 21 attack. <https://goo.gl/nGpNw3>. Accessed: July 17, 2018.
- [Eddy, 2007] Eddy, W. (2007). *TCP SYN Flooding Attacks and Common Mitigations*. RFC 4987, RFC Editor.
- [eEye Digital Security, 2001] eEye Digital Security (2001). Analysis: .ida "code red" worm. <https://goo.gl/qiFP14>. Accessed: July 16, 2018.
- [ENISA, 2016] ENISA (2016). DDoS on DNS root servers. <https://www.enisa.europa.eu/publications/info-notes/ddos-on-dns-root-servers>. Accessed: July 17, 2018.
- [eSecurity Planet, 2013] eSecurity Planet (2013). Ddos attacks: Growing, but how much? <http://goo.gl/HhFkt>. Accessed: July 17, 2018.
- [Farinacci et al., 2013] Farinacci, D., Fuller, V., Meyer, D., & Lewis, D. (2013). *The Locator/ID Separation Protocol (LISP)*. RFC 6830, RFC Editor. <http://www.rfc-editor.org/rfc/rfc6830.txt>.
- [Farrell & Tschofenig, 2014] Farrell, S. & Tschofenig, H. (2014). *Pervasive Monitoring Is an Attack*. BCP 188, RFC Editor. <http://www.rfc-editor.org/rfc/rfc7258.txt>.
- [Fayaz et al., 2015] Fayaz, S. K., Tobioka, Y., Sekar, V., & Bailey, M. (2015). Bohatei: Flexible and elastic ddos defense. In *24th USENIX Security Symposium (USENIX Security 15)* (pp. 817–832). Washington, D.C.: USENIX Association.

- [Ford et al., 2013] Ford, A., Raiciu, C., Handley, M., & Bonaventure, O. (2013). *TCP Extensions for Multipath Operation with Multiple Addresses*. RFC 6824, RFC Editor. <http://www.rfc-editor.org/rfc/rfc6824.txt>.
- [Freier et al., 2011] Freier, A., Karlton, P., & Kocher, P. (2011). *The Secure Sockets Layer (SSL) Protocol Version 3.0*. RFC 6101, RFC Editor. <http://www.rfc-editor.org/rfc/rfc6101.txt>.
- [Geng et al., 2002] Geng, X., Huang, Y., & Whinston, A. B. (2002). Defending wireless infrastructure against the challenge of ddos attacks. *Mobile Networks and Applications*, 7(3), 213–223.
- [Gont, 2013] Gont, F. (2013). *Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery*. RFC 6980, RFC Editor.
- [GSA Store,] GSA Store. GSA Advantage. <http://1.usa.gov/1ggEgFN>. [Online; accessed 19-Dec-2015].
- [Handley & Greenhalgh, 2004] Handley, M. & Greenhalgh, A. (2004). Steps towards a dos-resistant internet architecture. In *Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture*, FDNA '04 (pp. 49–56). New York, NY, USA: ACM.
- [Hughes, 1992] Hughes, G. (1992). Computer crime: Implications of recent english decisions. *Computers and Law: Journal for the Australian and New Zealand Societies for Computers and the Law*, 40, 23–25.
- [Imperva, 2015] Imperva (2015). The top 10 ddos attack trends.
- [Ioannidis & Bellovin, 2002] Ioannidis, J. & Bellovin, M. S. (2002). Implementing pushback: Router-based defense against ddos attacks.
- [Johnson et al., 2004] Johnson, D., Perkins, C., & Arkko, J. (2004). *Mobility Support in IPv6*. RFC 3775, IETF.
- [Jyothi et al., 2016] Jyothi, V., Wang, X., Addepalli, S. K., & Karri, R. (2016). Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect ddos attacks. In *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)* (pp. 587–588).
- [Karat, 2016] Karat, S. (2016). The top 5 ddos attack types we saw in 2015. <https://goo.gl/rVYJGN>. Accessed: July 16, 2018.
- [Kaspersky Labs,] Kaspersky Labs. DDoS attacks in Q1 2018. <https://securelist.com/ddos-report-in-q1-2018/85373/>. [Online; accessed 21-Aug-2018].

- [Kent & Atkinson, 1998] Kent, S. & Atkinson, R. (1998). *Security Architecture for the Internet Protocol*. RFC 2401, RFC Editor.
- [Khare et al., 2010] Khare, V., Jen, D., Zhao, X., Liu, Y., Massey, D., Wang, L., Zhang, B., & Zhang, L. (2010). Evolution towards global routing scalability. *IEEE Journal on Selected Areas in Communications*, 28(8), 1363–1375.
- [Krebs, 2016] Krebs, B. (2016). Krebsonsecurity hit with record ddos. <http://goo.gl/HhFkt>. Accessed: July 17, 2018.
- [Kupreev; & Badovskaya, 2018] Kupreev, A. K. O. & Badovskaya, E. (2018). Ddos attacks in q1 2018. <https://securelist.com/ddos-report-in-q1-2018/85373/>. Accessed: July 17, 2018.
- [Kwan et al., 1995] Kwan, T. T., McGrath, R. E., & Reed, D. A. (1995). Ncsa's world wide web server: design and performance. *Computer*, 28(11), 68–74.
- [Leech et al., 1996] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., & Jones, L. (1996). *SOCKS Protocol Version 5*. RFC 1928, RFC Editor.
- [Lemon, 2002] Lemon, J. (2002). Resisting SYN flood DoS attacks with a SYN cache. In *Proceedings of the BSD Conference 2002 on BSD Conference*, BSDC'02 (pp. 10–10). Berkeley, CA, USA: USENIX Association.
- [Li, 2011] Li, T. (2011). *Recommendation for a Routing Architecture*. RFC 6115, IRTF.
- [Liu et al., 2011] Liu, H., Sun, Y., Valgenti, V. C., & Kim, M. S. (2011). Trustguard: A flow-level reputation-based ddos defense system. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)* (pp. 287–291).
- [Liu et al., 2016] Liu, W., Qu, W., Gong, J., & Li, K. (2016). Detection of superpoints using a vector bloom filter. *IEEE Transactions on Information Forensics and Security*, 11(3), 514–527.
- [Liu et al., 2010] Liu, X., Yang, X., & Xia, Y. (2010). Netfence: Preventing internet denial of service from inside out. *SIGCOMM Comput. Commun. Rev.*, 40(4), 255–266.
- [Liyanage et al., 2015] Liyanage, M., Ahmed, I., Ylianttila, M., Santos, J. L., Kantola, R., Perez, O. L., Itzazelaia, M. U., d. Oca, E. M., Valtierra, A., & Jimenez, C. (2015). Security for future software defined mobile networks. In *Next Generation Mobile Applications, Services and Technologies, 2015 9th International Conference on* (pp. 256–264).

- [LKDDb, 2018] LKDDb (2018). Tcp syn cookie support. https://cateee.net/lkddb/web-lkddb/SYN_COOKIES.html. Accessed: July 10, 2018.
- [Lua et al., 2014] Lua, R. P., Wah, C. H., & Ng, W. K. (2014). Cornstarch effect: intensifying flow resistance for increasing ddos attacks in autonomous overlays. In *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)* (pp. 537–538).
- [Luo et al., 2013] Luo, H., Lin, Y., Zhang, H., & Zukerman, M. (2013). Preventing ddos attacks by identifier/locator separation. *IEEE Network*, 27(6), 60–65.
- [M. Handley, 2006] M. Handley, E. R. (2006). *Internet Denial-of-Service Considerations*. RFC 4732, RFC Editor.
- [Mahadev et al., 2016] Mahadev, Kumar, V., & Kumar, K. (2016). Classification of ddos attack tools and its handling techniques and strategy at application layer. In *2016 2nd International Conference on Advances in Computing, Communication, Automation (ICACCA) (Fall)* (pp. 1–6).
- [Marnerides & Mauthe, 2016] Marnerides, A. K. & Mauthe, A. U. (2016). Analysis and characterisation of botnet scan traffic. In *2016 International Conference on Computing, Networking and Communications (ICNC)* (pp. 1–7).
- [McGann & Malone, 2006] McGann, O. & Malone, D. (2006). Flow label filtering feasibility. In A. Blyth (Ed.), *EC2ND 2005* (pp. 41–49). London: Springer London.
- [Mockapetris, 1983] Mockapetris, P. (1983). *Domain names: Concepts and facilities*. RFC 882, RFC Editor. <http://www.rfc-editor.org/rfc/rfc882.txt>.
- [Mockapetris, 1987] Mockapetris, P. (1987). *Domain names - implementation and specification*. STD 13, RFC Editor. <http://www.rfc-editor.org/rfc/rfc1035.txt>.
- [Moskowitz et al., 2015] Moskowitz, R., Heer, T., Jokela, P., & Henderson, T. (2015). *Host Identity Protocol Version 2 (HIPv2)*. RFC 7401, RFC Editor. <http://www.rfc-editor.org/rfc/rfc7401.txt>.
- [Moskowitz & Nikander, 2006] Moskowitz, R. & Nikander, P. (2006). *Host Identity Protocol (HIP) Architecture*. RFC 4423, RFC Editor.
- [Nagami et al., 2007] Nagami, K., Uda, S., Ogashiwa, N., Esaki, H., Wakikawa, R., & Ohnishi, H. (2007). *Multi-homing for small scale fixed network Using Mobile IP and NEMO*. RFC 4908, RFC Editor.

- [Nazario & Holz, 2008] Nazario, J. & Holz, T. (2008). As the net churns: Fast-flux botnet observations. In *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)* (pp. 24–31).
- [Nezhad et al., 2016] Nezhad, S. M. T., Nazari, M., & Gharavol, E. A. (2016). A novel dos and ddos attacks detection algorithm using arima time series model and chaotic system in computer networks. *IEEE Communications Letters*, 20(4), 700–703.
- [(NIST), 2001] (NIST), R. K. (2001). *PBX Vulnerability Analysis: Finding Holes in Your PBX Before Someone Else Does*. Publication SP 800-24, Technology Administration U.S. Department of Commerce.
- [OED, 2018] OED (2018). *Oxford English Dictionary*. Oxford University Press. Accessed: July 10, 2018.
- [Park, 2003] Park, K. (2003). Scalable ddos protection using route-based filtering - discex iii demonstration. In *Proceedings DARPA Information Survivability Conference and Exposition*, volume 2 (pp. 97 vol.2–).
- [Parno et al., 2007] Parno, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., & Hu, Y.-C. (2007). Portcullis: Protecting connection setup from denial-of-capability attacks. *SIGCOMM Comput. Commun. Rev.*, 37(4), 289–300.
- [Peraković et al., 2016] Peraković, D., Periša, M., Cvitić, I., & Husnjak, S. (2016). Artificial neuron network implementation in detection and classification of ddos traffic. In *2016 24th Telecommunications Forum (TELFOR)* (pp. 1–4).
- [Phoomikiattisak, 2016] Phoomikiattisak, D. (2016). *Mobility as first class functionality : ILNPv6 in the Linux kernel*. dissertation, University of St Andrews.
- [Phoomikiattisak & Bhatti, 2015] Phoomikiattisak, D. & Bhatti, S. N. (2015). Mobility as a first class function. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 850–859).
- [Radware, 2003] Radware (2003). Today’s fbi, facts and figures. <https://goo.gl/8Kg3R7>. Accessed: July 17, 2018.
- [Radware, 2017] Radware (2017). History of ddos attacks. <https://goo.gl/k3Jnhj>. Accessed: July 17, 2018.
- [Ranjan et al., 2009] Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., & Knightly, E. (2009). Ddos-shield: Ddos-resilient scheduling to counter application layer attacks. *IEEE/ACM Transactions on Networking*, 17(1), 26–39.

- [Rao et al., 2010] Rao, S., Wang, Y., & I. Tao, X. (2010). The comprehensive trust model in p2p based on improved eigentrust algorithm. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 3 (pp. 822–825).
- [Rootops, 2015] Rootops (2015). Events of 2015-11-30. <http://root-servers.org/news/events-of-20151130.txt>. Accessed: July 17, 2018.
- [Santanna et al., 2017] Santanna, J. J., d. O. Schmidt, R., Tuncer, D., Sperotto, A., Granville, L. Z., & Pras, A. (2017). Quiet dogs can bite: Which booters should we go after, and what are our mitigation options? *IEEE Communications Magazine*, 55(7), 50–56.
- [Savage et al., 2001] Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2001). Network support for ip traceback. *IEEE/ACM Transactions on Networking*, 9(3), 226–237.
- [Scholtz, 1982] Scholtz, R. (1982). The origins of spread-spectrum communications. *IEEE Transactions on Communications*, 30(5), 822–854.
- [Shevtekar & Ansari, 2007] Shevtekar, A. & Ansari, N. (2007). A proactive test based differentiation technique to mitigate low rate dos attacks. In *2007 16th International Conference on Computer Communications and Networks* (pp. 639–644).
- [Shue et al., 2012] Shue, C. A., Kalafut, A. J., Allman, M., & Taylor, C. R. (2012). On building inexpensive network capabilities. *SIGCOMM Comput. Commun. Rev.*, 42(2), 72–79.
- [Sikdar et al., 2001] Sikdar, B., Kalyanaraman, S., & Vastola, K. S. (2001). Analytic models and comparative study of the latency and steady-state throughput of tcp tahoe, reno and sack. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, volume 3 (pp. 1781–1787 vol.3).
- [Simpson, 2016] Simpson, B. (2016). *Multihoming with ILNP in FreeBSD*. dissertation, University of St Andrews.
- [Sisalem et al., 2009] Sisalem, D., Floroiu, J., Kuthan, J., Abend, U., & Schulzrinne, H. (2009). *Denial of Service Attacks on VoIP and IMS Services*, (pp. 350–). Wiley Telecom.
- [Stoica et al., 2002] Stoica, I., Adkins, D., Zhuang, S., Shenker, S., & Surana, S. (2002). Internet indirection infrastructure. *SIGCOMM Comput. Commun. Rev.*, 32(4), 73–86.

- [Stolfo et al., 2005] Stolfo, S. J., Hershkop, S., Bui, L. H., Ferster, R., & Wang, K. (2005). Anomaly detection in computer security and an application to file system accesses. In M.-S. Hacid, N. V. Murray, Z. W. Raś, & S. Tsumoto (Eds.), *Foundations of Intelligent Systems* (pp. 14–28). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [T. Rontti, 2012] T. Rontti, A. M. Juuso, A. T. (2012). Preventing DoS attacks in NGN networks with proactive specification-based fuzzing. *IEEE Communications Magazine*, 50(9), 164–170.
- [The Hacker News, 2018] The Hacker News (2018). Biggest-ever ddos attack (1.35 tbps) hits github website. <https://goo.gl/NTNvsU>. Accessed: July 17, 2018.
- [Trotter, 2001] Trotter, G. (2001). *Terminology for Forwarding Information Base (FIB) based Router Performance*. RFC 3222, RFC Editor.
- [Tseung et al., 2017] Tseung, C. Y., Chow, K. P., & Zhang, X. (2017). Extended abstract: Anti-ddos technique using self-learning bloom filter. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)* (pp. 204–204).
- [Verisign, 2018] Verisign (2018). Q1 2018 verisign distributed denial of service trends report. https://www.verisign.com/assets/report-ddos-trends-Q12018_en_GB.pdf. Accessed: July 16, 2018.
- [Vixie et al., 1997] Vixie, P., Thomson, S., Rekhter, Y., & Bound, J. (1997). *Dynamic Updates in the Domain Name System (DNS UPDATE)*. RFC 2136, RFC Editor. <http://www.rfc-editor.org/rfc/rfc2136.txt>.
- [Wang et al., 2015] Wang, A., Mohaisen, A., Chang, W., & Chen, S. (2015). Delving into internet ddos attacks by botnets: Characterization and analysis. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (pp. 379–390).
- [Wang et al., 2010] Wang, Y., Bi, J., & Wu, J. (2010). Nol: Name overlay service for improving internet routing scalability. In *2010 Second International Conference on Advances in Future Internet* (pp. 17–21).
- [Warrender et al., 1999] Warrender, C., Forrest, S., & Pearlmutter, B. (1999). Detecting intrusions using system calls: alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344)* (pp. 133–145).
- [Wellington, 2000] Wellington, B. (2000). *Secure Domain Name System (DNS) Dynamic Update*. RFC 3007, RFC Editor.

- [Wong et al., 2004] Wong, C., Bielski, S., McCune, J. M., & Wang, C. (2004). A study of mass-mailing worms. In *Proceedings of the 2004 ACM Workshop on Rapid Malcode*, WORM '04 (pp. 1–10). New York, NY, USA: ACM.
- [Wood & Stankovic, 2004] Wood, A. D. & Stankovic, J. A. (2004). A taxonomy for denial-of-service attacks in wireless sensor networks.
- [Xiaohu Xu, 2010] Xiaohu Xu (2010). *Routing Architecture for the Next Generation Internet (RANGI)*. Internet-Draft draft-xu-rangi-04, IETF Secretariat. <http://www.ietf.org/internet-drafts/draft-xu-rangi-04.txt>.
- [Yaar et al., 2004] Yaar, A., Perrig, A., & Song, D. (2004). Siff: a stateless internet flow filter to mitigate ddos flooding attacks. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004* (pp. 130–143).
- [Yang et al., 2013] Yang, J., Park, M., & Chung, T. (2013). A study on low-rate ddos attacks in real networks. In *2013 International Conference on Information Science and Applications (ICISA)* (pp. 1–4).
- [Yang et al., 2005] Yang, X., Wetherall, D., & Anderson, T. (2005). A dos-limiting network architecture. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '05 (pp. 241–252). New York, NY, USA: ACM.
- [Yang et al., 2008] Yang, X., Wetherall, D., & Anderson, T. (2008). Tva: A dos-limiting network architecture. *IEEE/ACM Trans. Netw.*, 16(6), 1267–1280.
- [Ylmaz et al., 2018] Ylmaz, E. N., Ciylan, B., Gonen, S., Sindiren, E., & Karacaylmaz, G. (2018). Cyber security in industrial control systems: Analysis of dos attacks against plcs and the insider effect. In *2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)* (pp. 81–85).
- [Ylonen & Lonvick, 2006] Ylonen, T. & Lonvick, C. (2006). *The Secure Shell (SSH) Transport Layer Protocol*. RFC 4253, RFC Editor. <http://www.rfc-editor.org/rfc/rfc4253.txt>.
- [Z. Hu et al., 2014] Z. Hu, Z., Zhu, L., Ardi, C., Katz-Bassett, E., Madhyastha, H. V., Heidemann, J., & Yu, M. (2014). The need for end-to-end evaluation of cloud availability. In M. Faloutsos & A. Kuzmanovic (Eds.), *Passive and Active Measurement* (pp. 119–130). Cham: Springer International Publishing.

- [Zargar et al., 2009] Zargar, S., Weiss, M., Caicedo, C., & Joshi, J. (2009). Security in dynamic spectrum access systems: A survey. In *Telecommunications Policy Research Conference*.
- [Zargar et al., 2013] Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE Communications Surveys Tutorials*, 15(4), 2046–2069.
- [Zhang et al., 2004] Zhang, X.-F., Sun, Y.-F., & Zhao, Q.-S. (2004). Intrusion detection based on sub-set of system calls. 32, 1338–1341.
- [Zhang et al., 2009] Zhang, Y., Wan, Z., & Wu, M. (2009). An active ddos defense model based on packet marking. In *2009 Second International Workshop on Computer Science and Engineering*, volume 1 (pp. 435–438).
- [Zhou et al., 2010] Zhou, T., Wang, X.-F., Feng, L., & Wang, J. (2010). Research on host-level security situational awareness. *2010 3rd International Conference on Computer Science and Information Technology*, 1, 575–579.
- [Zhuang et al., 2013] Zhuang, R., Zhang, S., Bardas, A., DeLoach, S. A., Ou, X., & Singhal, A. (2013). Investigating the application of moving target defenses to network security. In *2013 6th International Symposium on Resilient Control Systems (ISRCS)* (pp. 162–169).
- [Zilberman et al., 2017] Zilberman, P., Puzis, R., & Elovici, Y. (2017). On network footprint of traffic inspection and filtering at global scrubbing centers. *IEEE Transactions on Dependable and Secure Computing*, 14(5), 521–534.